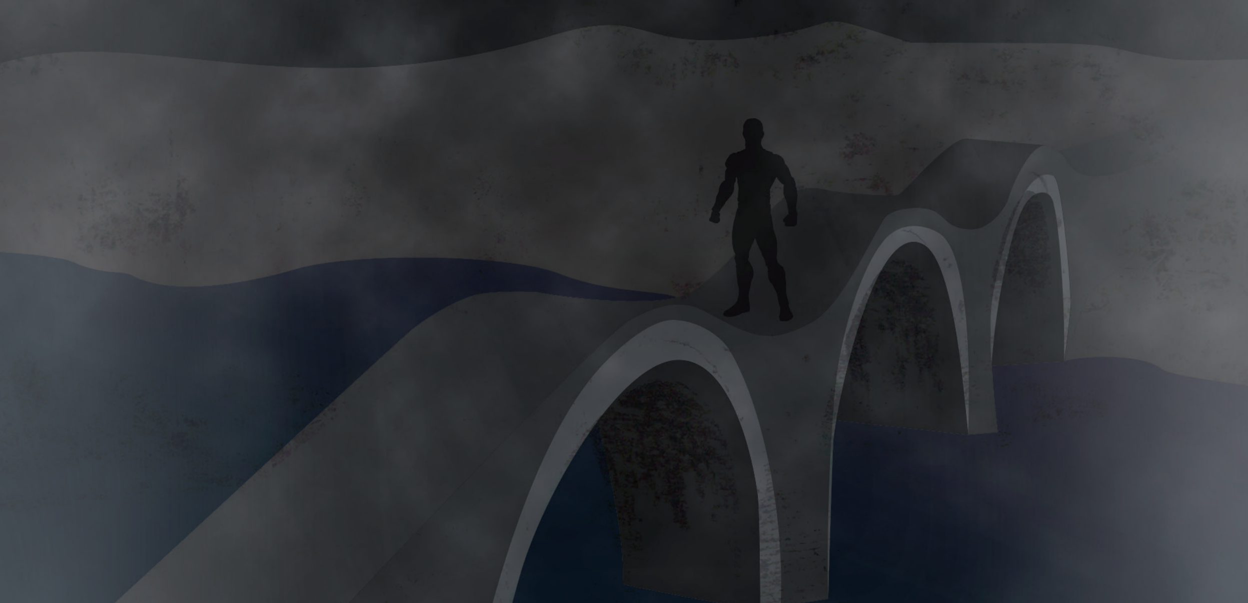
all rights reversed

HACKER

Δεκέμβρης 2013 # Τιμή τεύχους 4€

τεύχος 027 - Mind The Gap Edition

ΓΕΦΥΡΕΣ για αποτελεσματικές επιθέσεις MiTM [*]



Γράψτε τα πρώτα σας web-apps, σε PHP

Πώς θα φτιάξετε έναν πραγματικά ξεχωριστό router

Επιθέσεις Cross-Site Request Forgery, αλλά για αρχάριους

[*] Και δεν πειράζει που ως τώρα αγνοούσατε το bridging

all rights reversed

HACKER

tefhos 27o

Περιεχόμενα

Εκδότης / Διευθυντής	Χρήστος Βαρελάς
Αρχισυντάκτης	Παναγιώτης Βαρελάς
Σύμβουλοι έκδοσης	Νίκος Μουρατίδης, Βούλα Παυλίδου
Σύνταξη	Παναγιώτης Βαρελάς, Χρήστος Βαρελάς, Ανδρέας Βενιέρης, Γιώργος Γιάννου, Μανώλης Κιαγιάς, Ιωάννης Κονιάρης, Πέτρος Κυλαδίτης, Γιώργος Μελέκος, Νίκος Μουρατίδης, Νικόλαος Σηφάκης, Μέσος Παπαδόπουλος (ΤΜ), Χρίστος Τόμπρας
Γραμματική επιμέλεια	Βούλα Παυλίδου
Καλλιτεχνική Επιμέλεια	Πέτρος Φιλιππίδης

Το ηλεκτρονικό περιοδικό deltaHacker είναι συνδρομητικό και εκδίδεται από την Parabing Creations δώδεκα φορές το χρόνο, κάθε μήνα. Η Parabing Creations έχει την έδρα της στη Μαιώτιδος 3, 55133, Καλαμαριά. Ο δικτυακός τόπος του περιοδικού είναι στο http://deltahacker.gr. Το email επικοινωνίας είναι το talk2us@deltahacker.gr και ειδικά για τις συνδρομές είναι το subscriptions@deltahacker.gr. Οι γενικές πληροφορίες για τις συνδρομές παρατίθενται στο http://deltahacker.gr/subscriptions και οι ίδιες οι παραγγελίες γίνονται από το http://deltahacker.gr/pdf12order. Όλες οι απόψεις που εκφράζονται στα άρθρα δεν εκφράζουν απαραίτητα και τη γνώμη του περιοδικού.

(*) deltaCast s01e12 | Armitage Hacking
http://deltahacker.gr/deltacast-s01e12

Η μόνη διέξοδος, για κάθε ανεξάρτητη ιδέα!

Μέσα στο 2013 έγινε κάτι συντ... Συγγνώμη, βιάζομαι. Εύχομαι σε όλους καλή χρονιά, και είθε η τεχνολογία να είναι προσιτή και χρήσιμη για τις ανάγκες μας.

by @arkoudos

Δεν θα έπρεπε να είμαι αγενής, το ξέρω. Πρώτα εύχεται ο κόσμος — κι έτσι άλλωστε τελείωσε και το editorial μου, στο deltaHacker 026. Αλλά, όπως έλεγα και πριν, το 2013 έγινε κάτι συνταρακτικό, κάτι που, εγώ τουλάχιστον, δεν είχα υπολογίσει ότι θα γινόταν, όχι τόσο σύντομα, όχι τόσο εντυπωσιακά. Καταλαβαίνετε, λοιπόν, ότι βιάζομαι λίγο να μοιραστώ μαζί σας τον ενθουσιασμό μου.

Δεν αναφέρομαι στις γραμμές VDSL, στα smartphones ή στα tablets που έχουν κατακλύσει τον τόπο κ.λπ. Αφορά στην τεχνολογία ο ενθουσιασμός μου, ωστόσο αναφέρομαι στην μετάβαση της ενημέρωσης από την ΕΡΤ, δηλαδή από την τηλεόραση, στο Διαδίκτυο.

Όπως όλες οι μεγάλες γέννες, βέβαια, έτσι κι αυτή ήρθε βίαια, με δάκρυα και πολύ πόνο: Χρειάστηκε μια απαράδεκτη, προσωπικά πιστεύω, κυβερνητική μεθόδευση, μπόλικη δόση υποκρισίας, μια δόση προσωπικού μεγαλείου, άθλιο παρελθόν πολλών ετών, μια τζούρα Νέρωνα, και μια αψυχολόγητη απόφαση.

Όλα αυτά σκότωσαν την συμβατικής μετάδοσης Ελληνική Ραδιοφωνία και Τηλεόραση, και γέννησαν άλλη μία ελπίδα για το Διαδίκτυο — την πιο μεγάλη μέχρι τώρα, τουλάχιστον κατ' εμέ. Γιατί ελπίδα; Εξηγούμαι.

Όταν η κυβέρνηση κατέβασε τα ρολά, οι εργαζόμενοι ένιωσαν την ανάγκη να μιλήσουν — για πρώτη φορά, ίσως— για τον εαυτό τους. Κι αυτό *δεν* μπορεί να γίνει από τις διόδους των παραδοσιακών ΜΜΕ — όχι τουλάχιστον τόσο εύκολα. Επέλεξαν έτσι, από ανάγκη, το μόνο διαθέσιμο μέσο: το Διαδίκτυο.

Η ανταπόκριση του κόσμου ήταν πρωτοφανής. Στα δικά μου, αθώα μάτια, έμοιαζε —ίσως υπερβολικά— με ένα "δικτυακό Σύνταγμα". Εκατομμύρια κόσμου, κάθε ηλικίας, χωρίς απαραιτήτως προηγούμενη σχέση με το μέσο, έμαθαν ότι η "ελεύθερη" ΕΡΤ εκπέμπει από τη μικρή οθόνη του υπολογιστή, με URLs που άλλαζαν κάθε λίγο, κι όχι πια από την μεγάλη οθόνη του παραδοσιακού κουτιού. Ενεπλάκη λοιπόν ο κόσμος σε μια διαδικασία που θύμιζε ενοχλητικά το "εδώ πολυτεχνείο" —δεν επιχειρώ καμία σύνδεση από πλευράς περιεχομένου, φυσικά— μόνο που τώρα δεν μιλούσαμε για πειρατικό ραδιόφωνο, αλλά για απλή, δικτυακή μετάδοση σήματος.

Η δε μεγάλη οθόνη της τηλεόρασης δεν έκρυψε τα δόντια της "συμπαράστασής" της. Πράγματι, παρά την "πειρατική" εκπομπή των εργαζομένων της ΕΡΤ, η οποία θα μπορούσε να αποτελέσει είδηση —για να μην πω υλικό συμπαράστασης μεταξύ συναδέλφων— όχι μόνο εφαρμόστηκε απόλυτη σιγή ιχθύος, αλλά κι όταν το ...κομμουνιστικότατο, πάλαι ποτέ 902 TV, ξεκίνησε την αναμετάδοση του δικτυακού σήματος της ΕΡΤ, άρχισε ένα άνευ προηγουμένου, απίστευτο παιχνίδι "τώρα μπορείς να παίξεις — τώρα σε λογοκρίνω" με την Digea. Ξέρετε καλά από πότε είχαμε να δούμε τόσο "ελεύθερη τηλεόραση", έτσι δεν είναι; [*]

Φανταστείτε: Οι εργαζόμενοι στον Ελληνικό, κολοσσιαίο τηλεοπτικό φορέα, να βασίζονται τελικά σε μια εταιρεία που μάλλον ανερχόμενη θα την έλεγες, την BitsnBytes [**], προκειμένου να ενημερώσει το ασύλληπτο πλήθος κόσμου που ήθελε ν' ακούσει μια διαφορετική

άποψη απ' αυτή του κυβερνητικού εκπροσώπου (την τωρινή, όχι του 2012, που όσο να 'ναι ήταν κομμάτι διαφορετική).

Και υπενθυμίζω ότι η εναλλακτική άποψη κατάφερνε να ακουστεί παρά τις επιλογές εκείνων που ελέγχουν τη συμβατική ροή ενημέρωσης, καθώς κι εκείνων που απροκάλυπτα λογοκρίνουν το τηλεοπτικό σήμα, για ξεκάθαρα πολιτικούς λόγους.

Θα μου επιτρέψετε να πω ότι κάποιες μάσκες έπεσαν το 2013 — και μάλιστα προκάλεσαν μεγάλο θόρυβο.

Αλλά δεν ήταν μόνο η υπόθεση της ΕΡΤ, που κατέδειξε ότι υπάρχουν και οι εναλλακτικές πηγές ενημέρωσης. Το OmniaTV.com, για παράδειγμα, συνεχίζει να εκπέμπει εικόνα, το Radiobubble.gr κάνει ακόμα τις ραδιοφωνικές του εκπομπές, ενώ το ThePressProject.gr παρέχει μια αξιόλογης ποιότητας δικτυακή επιλογή από streams.

Όλα αυτά, όσο κλισέ κι αν δείχνει αυτό που θα γράψω, δεν θα γίνονταν χωρίς τις δικές σας δυνάμεις. Τα μέσα που προανέφερα, μαζί με τα παραδοσιακής μορφής —αλλά, σας διαβεβαιώνω, όχι παραδοσιακά παραγόμενα— Unfollow, HotDoc, deltaHacker (και deltaCast) [***], έχουν την απόλυτη ανάγκη της στήριξής σας.

Χωρίς πρόσβαση στις παραδοσιακές πηγές χρηματοδότησης, τα περισσότερα αντισυμβατικά μέσα έχουν ανάγκη της δικής μας χορηγίας, ώστε να συνεχίσουν να υφίστανται και να δημιουργούν και να εκφράζουν γνώμη ελεύθερα — κι όχι όπως θέλουν άλλοι.

Δεν κρίνω τα αποτελέσματα. Σε άλλους αρέσουν, σε άλλους όχι. Άλλοι, π.χ., τα βρίσκουν αμιγώς επαναστατικά, άλλοι θεωρούν ότι αποτελούν τη δικτυακή μορφή του λαϊκισμού. Δεν είναι ανάγκη να συμφωνήσετε καν μαζί μου για τα αίτια πίσω από την έλευσή τους. Ίσως να σκέφτεστε ότι όλα ξεκίνησαν από μια ανάγκη για κάθαρση, ίσως να υποψιάζεστε ότι τα κίνητρα ήταν η επιθυμία για ενημέρωση χωρίς λογοκρισία, ίσως να υποψιάζεστε και κάτι άλλο: Σε κάθε περίπτωση, δεν χρειάζεται να συμφωνούμε.

Εγώ κρατάω —και μοιράζομαι μαζί σας— τα εξής: Τη στιγμή που τα ελεγχόμενα μέσα (από κυβερνήσεις, διαφημιζόμενους, παραδοσιακές αντιλήψεις) πίστευαν ότι έχουν τον απόλυτο έλεγχο, η μόνη διέξοδος για κάθε ανεξάρτητη ιδέα φάνηκε να είναι η τεχνολογία και το Διαδίκτυο.

Αυτό σύντομα θα αλλάξει, μάλλον. Τόση ελευθερία δεν περιμένω να παραμείνει για πολύ ανεκτή. Θα θυμόσαστε ή θα έχετε μάθει τι έγινε στο παρελθόν, με το πειρατικό ραδιόφωνο. Αλλά όσο μπορούμε και προλαβαίνουμε να το απολαμβάνουμε μέσα το 2014, ας το απολαμβάνουμε. Τι λέτε κι εσείς; Και κυρίως, ας αντιληφθούμε *γιατί* δημιουργήθηκε αυτή η ανάγκη για εναλλακτικές πηγές ενημέρωσης. Είναι μάλλον το καλύτερο μάθημα που θα μπορούσαμε να έχουμε απ' αυτή την ιστορία.

========================

[*] Ίσως να θυμόσαστε το Alter και την Digea, που του κατέβαζε το σήμα σιγά σιγά: πρώτα την ποιότητα, μετά την εικόνα. Είχαμε δει, δηλαδή, τα προεόρτια πριν τη φετινή, ωμή παρέμβαση. Η ανύπαρκτη αντίδρασή μας, τότε, επέτρεψε εν πολλοίς να συμβούν αρκετά απ' όσα συνέβησαν το 2013.

[**] Η οποία BitsnBytes, ναι μεν είχε καταφέρει τα δύο άλλα, μεγάλα βήματα του διαδικτύου, τα ντοκυμαντέρ Debtocracy και Catastroika, ωστόσο κανείς δεν μπορούσε να φανταστεί ότι θα άντεχε στο χρόνο.

[***] Σ.τ.Ε. Από τα τέλη του περασμένου Απριλίου, το περιοδικό deltaHacker κυκλοφορεί σε αποκλειστικά ψηφιακή μορφή.



ουέλ μέλ

Αγαπητοί μου,

Χρονιάρες μέρες και πάλι, και λέω ν' αρχίσω την επιστολή μου με μια παραδοξολογία που την λέγαμε μικροί στο σχολείο για να δαιμονίσουμε, ούτως ειπείν, τον δάσκαλό μας των θρησκευτικών.

"Αν ο Θεός είναι παντοδύναμος", τον ρωτάγαμε, "μπορεί να φτιάξει έναν βράχο τόσο μεγάλο που να μην μπορεί να τον σηκώσει;"

Φυσικά δεν το είχαμε βγάλει εμείς, από κάπου το είχαμε ακούσει, αλλά για το εφέ έκανε μια χαρά. Ο θρησκευτικός μας πράγματι δαιμονιζόταν και πάσχιζε να μας απαντήσει είτε με φοβέρες για την ασέβειά μας, είτε με αμήχανους ταυτολογικούς συλλογιστικούς μαιάνδρους.

Η αδυναμία του θρησκευτικού μας μου φαίνεται σήμερα ελαφρώς περίεργη. Διότι η αναίρεση αυτής της παραδοξολογίας είναι σχετικώς απλή. Αυτό βέβαια το κατάλαβα αργότερα, δεν το ήξερα τότε που ήμασταν στο σχολείο.

Νομίζαμε, τότε, ότι είχαμε καταδείξει μια θεμελιώδη αντινομία στο δόγμα και δεν μας εξέπληττε το ότι ο θρησκευτικός μας δαιμονιζόταν. Μας φαινόταν αναμενόμενο.

Φευ... Η αλαζονεία της νεολαίας.

Δεν καταλαβαίναμε πως η αντινομία δεν βρισκόταν στο δόγμα αλλά στο ίδιο το ερώτημα.

Όταν αναρωτιέσαι για το αν μια οντότητα (ο Θεός εν προκειμένω) μπορεί ή δεν μπορεί να σηκώσει κάτι, είναι σαν να δέχεσαι ότι η οντότητα αυτή βρίσκεται στον ίδιο κόσμο με τον δικό σου, και ότι υπόκειται στους ίδιους με σένα περιορισμούς -π.χ. μάζα, βαρυτική έλξη, στρέβλωση χωροχρόνου κ.λπ. κ.λπ. Όμως αυτό το έχεις υποθέσει. Δεν το έχεις αποδείξει. Η σκέψη σου είναι κυκλική.

Πρώτα πρέπει να στηρίξεις όλες τις υποθέσεις σου, και μετά να κάνεις τον έξυπνο στον θρησκευτικό σου.

Αν το επιχειρήσεις, όμως, θα δεις ότι δεν έχει καν νόημα να προσπαθήσεις γιατί --εξ ορισμού-- ο Θεός εκλαμβάνεται ως ο δημιουργός του κόσμου τούτου και *όχι* ως ένας από τους κατοίκους του.

Το ερώτημά σου είναι άκυρο.

Είναι σαν να αναρωτιέσαι πώς και ο Όμηρος δεν είχε πέσει και αυτός θύμα του σαγηνευτικού τραγουδιού των Σειρήνων.

Το σκεφτόμουν, λοιπόν, το θέμα αυτό προ ημερών, που μιλούσαμε με την εξαδέλφη για τα Χρυσά Αβγά και για την ανάκαμψη, εκ νέου, των δημοσκοπικών ποσοστών τους.

Η θέση μου για το ζήτημα, αγαπητοί μου, έχει αποκρυσταλλωθεί. Η συγκεκριμένη οργάνωση πρέπει να απαγορευτεί. Απλά και σταράτα πράγματα. Πρέπει να απαγορευτεί εδώ και τώρα.

Αλλά --προσοχή σε αυτό-- να απαγορευτεί *ΟΧΙ* επειδή δεν αρέσουν

σε κάποιον οι ιδέες που πρεσβεύει. Ας πρεσβεύουν ό,τι θέλουν. Κουκιά τρώνε, κουκιά μαρτυράνε, που θα έλεγε και ο παππούς. Σ' αυτό δεν μπορώ να βοηθήσω.

Η οργάνωση πρέπει να απαγορευτεί γιατί είναι εγκληματική. Και είναι εγκληματική επειδή κάτω από τον μανδύα της μισαλλόδοξης και εθνικιστικής ρητορικής κρύβονται άνθρωποι που οπλίζονται, εκπαιδεύονται στρατιωτικά, χτυπάνε, μαχαιρώνουν, εκφοβίζουν, εκβιάζουν, δολοφονούν.

Αυτό είναι το εγκληματικό. Οι ιδέες τους είναι ηλίθιες. Με αφήνουν παγερά αδιάφορο. Τα όπλα τους όμως σκοτώνουν.

Η εξαδέλφη είχε αντιρρήσεις.

"Μα πώς μπορείς να ζητάς την απαγόρευσή τους όταν βλέπεις ότι είναι το τρίτο κόμμα σε δύναμη; Δεν σου ακούγεται αντιδημοκρατικό;"

"Αντιδημοκρατικό;" είπα εγώ. "Και δηλαδή ποιος σου είπε πως τα θεμέλια της δημοκρατίας είναι δημοκρατικά; Καθόλου δημοκρατικά δεν είναι."

Η εξαδέλφη παρέμεινε σιωπηλή.

"Κοίτα", συνέχισα, "πάρε για παράδειγμα τις εκλογές. Εκλογές γίνονται, είτε το θέλεις, είτε δεν το θέλεις. Κανείς δεν σε ρωτάει αν σ' αρέσει να γίνονται εκλογές και κανείς δεν πρόκειται να σε ρωτήσει. Εκλογές γίνονται γιατί το λέει το Σύνταγμα. Τέρμα. Το Σύνταγμα δεν το έχουμε βάλει εκεί για να το συζητάμε κάθε φορά. Αποφασίστηκε, ισχύει, πάει τέλειωσε. Να το αλλάξεις δε γίνεται με το έτσι θέλω, θέλει διαδικασίες, θέλει δουλειά. Και κάποια πράγματα δεν αλλάζουν με τίποτα."

"Ναι..." ξεκίνησε να λέει η εξαδέλφη επιφυλακτικά. "Αλλά εγώ δεν είπα να αλλάξουμε το Σύνταγμα... Εγώ έλεγα ότι δεν μπορείς να απαγορέψεις ένα κόμμα επειδή δεν συμφωνείς με αυτό. Είναι αντιδημοκρατικό..."

"Μα εδώ είναι το λάθος. Δεν θα το απαγορέψεις επειδή δεν συμφωνείς. Θα το απαγορέψεις επειδή κάτω από τον μανδύα του κοινοβουλευτισμού (και απολαμβάνοντας την προστασία του) αυτό το, τέλος πάντων, κόμμα δραστηριοποιείται σαν κρυφός στρατός που ετοιμάζεται για πόλεμο. *Αυτό* είναι το τεκμήριο της εγκληματικότητας. Και δεν νομιμοποιείται, όσοι μπούφοι και να τους ψηφίσουν, ό,τι ποσοστό και να μαζέψει στις εκλογές. Έτερον, εκάτερον."

"Μμμ..."

"Για να το πω πιο λιανά. Φαντάσου να γινόταν αύριο δημοψήφισμα με το ερώτημα αν θέλουμε να γίνονται δημοψηφίσματα στο μέλλον. Φαντάσου τώρα ότι κάπως μας ήρθε κι αποφασίσαμε όλοι σύσσωμοι ότι δεν θέλουμε. Τα βαρεθήκαμε. Τελεία. Δημοψηφίσματα τέρμα. Φαντάσου τώρα ότι περνάνε είκοσι χρόνια χωρίς δημοψηφίσματα και αρχίζουμε να το ξανασκεφτόμαστε. Ότι έχουν αρχίσει να μας ψιλολείπουν. Πώς θα το αποφασίσουμε να τα επαναφέρουμε, αν πλέον δεν μπορούμε να θέσουμε το ερώτημα σε δημοψήφισμα; Βλέπεις τι εννοώ; Βλέπεις την αντίφαση;"



"Βλέπω..."

Η οποία αντίφαση, αγαπητοί μου, είναι ίδιας τάξεως με την αντίφαση στην ερώτηση περί Θεού και βράχου που σας έγραφα παραπάνω. Ελπίζω ότι το βλέπετε.

Μ' αυτά και μ' αυτά, λοιπόν, περνούν οι μέρες και συνειδητοποιώ πως έχουμε αίφνης φτάσει στα μέσα Δεκεμβρίου.

Ποιος θα το φανταζόταν.

Είμαστε πλέον όλοι σε κέφι εορταστικό. Δέντρο δεν έχουμε στολίσει ακόμα, και δεν ξέρω αν θα στολίσουμε. Παίζεται.

Το θέμα ανέκυψε στο βραδινό μας τραπέζι, χτες Σάββατο, όπου τιμούσαμε τον ανιψιό, άρτι αφιχθέντα από την αλλοδαπή. Το ανακίνησε ο ίδιος, καθώς όσο απομακρύνεται από την παιδική ηλικία (δηλαδή: όσο πατάει η γάτα) τόσο τον πιάνει η νοσταλγία για τα αγνά εκείνα χρόνια (λέμε τώρα).

"Δέντρο δεν θα στολίσετε φέτος;" ρώτησε αθώα.

Ξέροντας τι θα ακολουθούσε έσκυψα σεμνά στο πιάτο μου, κι άρχισα να πηγαινοφέρνω με το πιρούνι μου τις πατάτες και το κουνουπίδι. "Άλλο πράγμα το αλού γκόμπι" είπα.

Με κοίταξαν όλοι άβολα.

"Εε... Ο συνδυασμός, εννοώ, πολύ ταιριαστός. Το σκόρδο, η πιπερόριζα, το κρεμμύδι και ο κόλιανδρος δένουν πολύ αρμονικά με τον κουρκουμά, τον ανθό μοσχοκάρυδου, και τον μάραθο".

Για λίγο επικράτησε ησυχία. Ακουγόταν μόνο ο ήχος μαχαιροπήρουνων σε πιάτα.

Την ησυχία διέκοψε η μητέρα της εξαδέλφης. "Όχι", είπε.

Συνέχισα να πηγαινοφέρνω με το πιρούνι μου τις πατάτες. Η Μαιρούλα μου έριξε ένα βλέμμα όλο ανησυχία και μετά στράφηκε προς κανέναν συγκεκριμένα. "Μπορώ να έχω λίγο νάαν;" ρώτησε προσεκτικά.

Η εξαδέλφη έσπρωξε την πιατέλα με τα νάαν προς την μεριά της Μαιρούλας.

"Δέντρο, *δεν* θα στολίσω φέτος", συνέχισε η μητέρα της εξαδέλφης. Με έμφαση στο "στολίσω", αν και δεν χρειαζόταν: Όλοι θυμόμασταν τις περσινές περιπέτειες με τα αγαλματίδια του Βισνού που λίγο έλειψαν να μετατρέψουν τη μικρή μας χριστουγεννιάτικη γωνιά σε ινδουιστικό ιερό.

Η μητέρα της εξαδέλφης είχε δηλώσει (τότε) ότι του χρόνου (φέτος) θα απέχει από τις σχετικές προετοιμασίες. Τέτοιοι που είμαστε.

Νομίζαμε ότι το είχε ξεχάσει. Αλλά πού να ξεχάσει.

Δέντρο λοιπόν φέτος γιοκ.

Μείναμε για λίγο σιωπηλοί. Αισθανόμασταν όλοι το βάρος των περιστάσεων.

"Λοιπόν, αίνιγμα", είπα εγώ για να αλλάξω θέμα.

"Πείτε μου. Αν ο Θεός είναι παντοδύναμος, μπορεί να φτιάξει ένα Χριστουγεννιάτικο δέντρο τόσο μεγάλο που να μην του φτάνει ο χρόνος να το στολίσει;"

"Ο Θεός δεν ασχολείται με βλακείες", είπε η μητέρα της εξαδέλφης. που είναι Ινδουίστρια.

Έπιασα ξανά το πιρούνι.

Σας ασπάζομαι,
Θείος Ακάκιος

Skill: Beginner
Tags: CSRF, GET, POST, nonce



by Ion

Πλαστογραφημένες δικτυακές αιτήσεις

Το Cross-Site Request Forgery δεν αποτελεί ιδιαίτερα γνωστή μέθοδο επίθεσης, αλλά να είσαστε σίγουροι ότι είναι αρκετά επικίνδυνη. Σε αυτό το άρθρο θα μάθουμε τι είναι και πώς υλοποιείται, στήνοντας το κατάλληλο δικτυακό περιβάλλον και πραγματοποιώντας μερικά ενδιαφέροντα πειράματα. Επιπρόσθετα, θα εξετάσουμε κι έναν απλό μηχανισμό άμυνας.

Το Cross-Site Request Forgery (ή απλά CSRF) αποτελεί μια επίθεση που πραγματοποιείται με στόχο κάποιο ευπαθές website. Ωστόσο το θύμα αυτής της επίθεσης δεν είναι το εκάστοτε website, ούτε ο server που το φιλοξενεί, ούτε ο διαχειριστής του. Το θύμα αυτής της επίθεσης είναι κάποιος από τους χρήστες του website! Η επίθεση CSRF προβλέπει την ύπουλη αποστολή δικτυακών αιτημάτων (requests) προς κάποιο website, χωρίς τη συναίνεση του χρήστη (επισκέπτη). Εκεί κολλάει και το "request forgery" , αφού τα αιτήματα μοιάζουν να προέρχονται από τον χρήστη, ενώ στην πραγματικότητα είναι κατασκευασμένα από τον επιτιθέμενο. Τα εν λόγω αιτήματα κρύβονται σε ιστοσελίδες που έχει επίσης κατασκευάσει ο επιτιθέμενος, ενώ παραπέμπουν σε άσχετα websites. Έτσι δικαιολογείται κι ο όρος "Cross-site", αφού η επίθεση λαμβάνει χώρα από έναν δικτυακό τόπο προς κάποιον άλλο. Όλα αυτά όμως είναι πολύ θεωρητικά. Ας δούμε για αρχή τι είναι αυτά τα δικτυακά αιτήματα και σε τι χρησιμεύουν, για να εξετάσουμε στη συνέχεια το πώς μπορούν να αξιοποιηθούν από τον επιτιθέμενο.

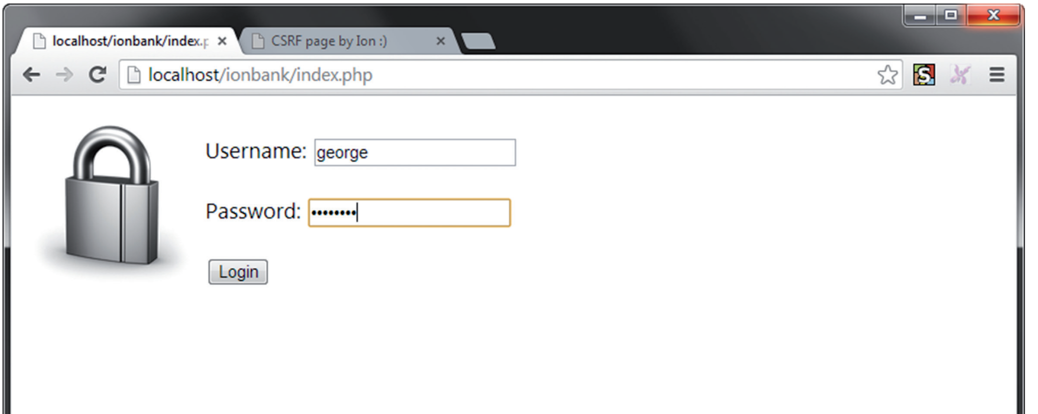
Αντικείμενα απάτης

Όταν χρησιμοποιούμε τον όρο request (αίτημα) αναφερόμαστε στις μεθόδους GET και POST, που προσφέρει το πρωτόκολλο HTTP. Οι μέθοδοι αυτές χρησιμοποιούνται σε όλα τα websites για τη μεταφορά δεδομένων από το χρήστη προς τον web server, για τη μεταφορά πληροφοριών από μία σελίδα σε μία άλλη, καθώς και για την ίδια τη μεταφορά των ιστοσελίδων. Για να δούμε οποιαδήποτε σελίδα ενός site, ο browser αποστέλλει ένα αίτημα GET στον web server και δηλώνει τη διεύθυνση της ζητούμενης σελίδας. Η απάντηση από τον web server περιλαμβάνει το περιεχόμενο κάποιας σελίδας HTML, η οποία ενδέχεται να περιέχει παραπομπές προς άλλα αρχεία (CSS, java script, εικόνες, φόρμες HTML που βρίσκονται σε ξεχωριστά αρχεία κ.ά). Για καθένα από αυτά τα αρχεία, ο browser στέλνει και πάλι ένα αίτημα GET, ζητώντας από τον web server το σχετικό περιεχόμενο. Όπως αναφέραμε προηγουμένως, η μέθοδος GET δεν χρησιμοποιείται μόνο για να προσπελάσουμε μια σελίδα, αλλά και για τη μεταφορά δεδομένων από σελίδα σε σελίδα. Χαρακτηριστικό παράδειγμα αποτελεί το λεγόμενο pagination. Πρόκειται για μια τεχνική που χωρίζει αυτόματα το περιεχόμενο σε ξεχωριστές σελίδες (π.χ., σε κάποιο δικτυακό κατάστημα, όπου τα προϊόντα μιας κατηγορίας εμφανίζονται σε ξεχωριστές σελίδες, ή σε κάποια μηχανή αναζήτησης, όπου τα αποτελέσματα της αναζήτησης χωρίζονται σε σελίδες). Όταν χρησιμοποιείται το pagination, τα URLs των επιμέρους σελίδων ενσωματώνουν στο τέλος τους διάφορες παραμέτρους (π.χ., "?page=24"). Ε, λοιπόν, αυτές οι παράμετροι αποτελούν δεδομένα που μεταφέρονται με τη μέθοδο GET. Βέβαια, η κύρια αποστολή της GET δεν είναι η μεταφορά παραμέτρων και δεδομένων, ασχέτως που αν χρησιμοποιείται πολύ συχνά για αυτό το σκοπό. Αυτή την αποστολή έχει η μέθοδος POST. Για παράδειγμα, όταν συμπληρώνουμε μια φόρμα HTML και πατάμε το κουμπί υποβολής, τα δεδομένα που δώσαμε μεταφέρονται μέσω της μεθόδου POST. Το πλεονέκτημα της μεθόδου POST έναντι της GET, είναι ότι τα δεδομένα που μεταφέρονται δεν τοποθετούνται στο URL και δεν είναι άμεσα ορατά.

Μέθοδος και προϋποθέσεις

Η αρχική περιγραφή και η λιγοστή θεωρία που ακολούθησε δεν αρκούν για να κατανοήσει κανείς τι είναι οι επιθέσεις CSRF. Για αυτό το λόγο, θα περιγράψουμε το πώς πραγματοποιείται μια τέτοια επίθεση — αναφέροντας ταυτόχρονα και τις όποιες προϋποθέσεις.

1. Ο χρήστης που αποτελεί το θύμα της επίθεσης, πρέπει είναι συνδεδεμένος στο website προς το οποίο θα πραγματοποιηθεί η επίθεση. Πρακτικά, αυτό σημαίνει ότι έχει κάνει login στο website και διατηρεί "ανοιχτή" τη σελίδα σε κάποια καρτέλα του browser — ή ότι έχει συνδεθεί στο παρελθόν και τώρα παραμένει συνδεδεμένος εξαιτίας των αποθηκευμένων cookies.
2. Ο κακόβουλος χρήστης εξαπατά τον χρήστη-θύμα και τον οδηγεί σε μια ειδική σελίδα, η οποία αποτελεί τμήμα του μηχανισμού επίθεσης. Η εξαπάτηση του θύματος πραγματοποιείται συνήθως με τη βοήθεια του social engineering ή με κάποια μέθοδο phishing. Για παράδειγμα, ο επιτιθέμενος μπορεί να στείλει στο θύμα ένα κατάλληλα γραμμένο και σχεδιασμένο email, το οποίο θα μοιάζει να προέρχεται από το Facebook ή κάποια άλλη γνωστή υπηρεσία. Σε αυτό το email, φυσικά, δεν θα υπάρχουν σύνδεσμοι (links) προς το αληθινό Facebook. Θα υπάρχουν links μόνο για τη σελίδα-παγίδα, στην οποία πρέπει να μεταβεί το θύμα.
3. Η σελίδα που έχει κατασκευάσει ο επιτιθέμενος περιέχει στον κώδικά της αιτήματα GET ή POST, προς το website στο οποίο είναι συνδεδεμένο το θύμα. Έτσι, όταν το θύμα φτάνει στη σελίδα του επιτιθέμενου, ο browser διαβάζει τον κώδικα HTML, συναντά τα εν λόγω αιτήματα και τα εκτελεί!

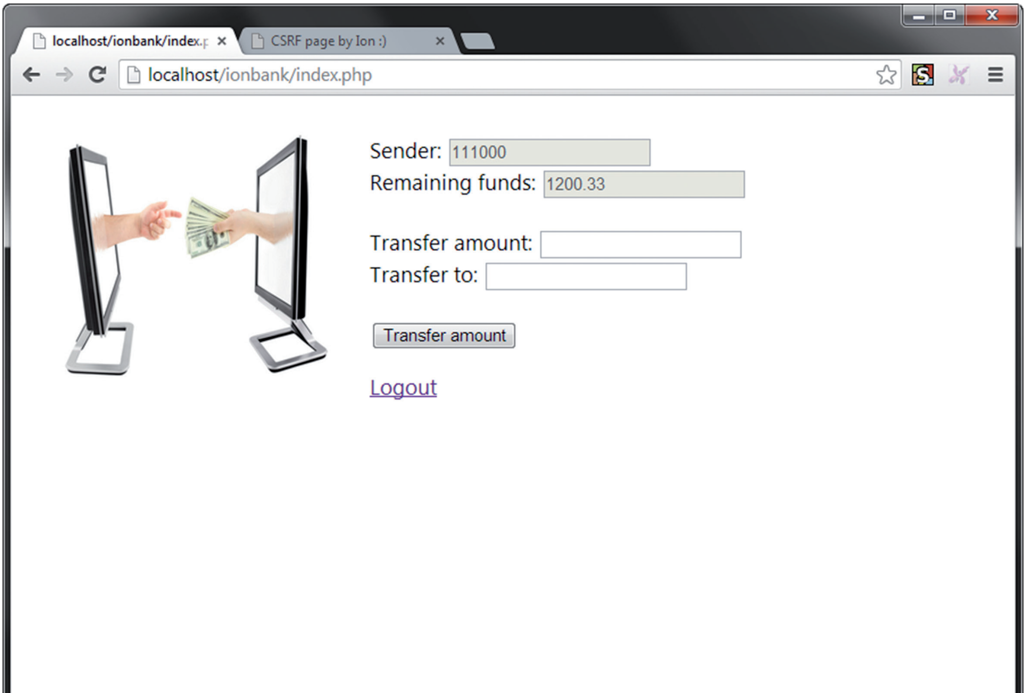


Το website της τράπεζας είναι εξαιρετικά απλό. Όταν το επισκεπτόμαστε, συναντάμε μόνο τη φόρμα σύνδεσης.

Τα αιτήματα που περιείχε η σελίδα του επιτιθέμενου και τα οποία εκτελεί ο browser του θύματος, μπορούν να αντλήσουν οποιαδήποτε πληροφορία ή και να πραγματοποιήσουν οποιαδήποτε ενέργεια (π.χ., να αλλάξουν το συνθηματικό του χρήστη, να τροποποιήσουν κάποια άλλη ρύθμιση, να διαγράψουν το λογαριασμό κ.ο.κ.). Δεν έχει καμία σημασία αν πρόκειται για δεδομένα ή ενέργειες που απαιτούν τη σύνδεση (login) του χρήστη, εφόσον τα αιτήματα φαίνεται να προέρχονται από τον browser ενός συνδεδεμένου χρήστη. Αν το website που γίνεται αποδέκτης των πλαστών αιτημάτων δεν πραγματοποιεί τον κατάλληλο έλεγχο εγκυρότητας, θα τα εκτελέσει κανονικά!

Από τη θεωρία στην πράξη

Η παραπάνω περιγραφή είναι πολύ πιο κατατοπιστική από όσα είπαμε προηγουμένως. Όμως, η αλήθεια είναι ότι δεν σταθήκαμε σε καμία λεπτομέρεια και, τέλος πάντων, δεν ξεφύγαμε από τη θεωρία. Στη συνέχεια, λοιπόν, θα εξετάσουμε μια επίθεση CSRF στην πράξη, κατασκευάζοντας το website που θα δεχτεί την επίθεση καθώς και την ειδική σελίδα του επιτιθέμενου, αυτή, με τα πλαστά αιτήματα. Για την κατασκευή όλων αυτών θα στηριχτούμε στον Apache web server και στη βάση δεδομένων MySQL. Συγκεκριμένα, θα χρησιμοποιήσουμε το πακέτο WAMP, που προσθέτει τα δύο παραπάνω προγράμματα μαζί με τον διερμηνέα (interpreter) της PHP και διάφορα βοηθητικά εργαλεία όπως το phpMyAdmin (`www.wampserver.com/en`).



Μετά τη σύνδεση στο τραπεζικό website, εμφανίζεται η φόρμα μεταφοράς χρημάτων. Η ίδια σελίδα εμφανίζει και το διαθέσιμο υπόλοιπο του λογαριασμού μας.

Στο σενάριό μας θα παίξουμε το ρόλο ενός κακόβουλου χάκερ, ο οποίος πραγματοποιεί επιθέσεις CSRF εναντίον του website μιας μεγάλης και γνωστής τράπεζας: της IonBank ;) Το website της τράπεζας περιλαμβάνει μια βάση δεδομένων με τους πελάτες. Για να μην πλατιάζουμε άσκοπα, θα υποθέσουμε ότι αυτή η βάση είναι εξαιρετικά απλή και για κάθε πελάτη διατηρεί μόνο τα ακόλουθα στοιχεία: Έναν κωδικό πελάτη, ένα username, ένα password και το σύνολο των χρημάτων. Όταν ένας πελάτης κάνει login στη σελίδα της τράπεζας, αποκτά πρόσβαση σε μία φόρμα μεταφοράς χρημάτων. Αυτή η φόρμα περιλαμβάνει ένα πεδίο για τη συμπλήρωση του χρηματικού ποσού που θα μεταφερθεί κι ένα πεδίο για τον κωδικό πελάτη που θα παραλάβει τα χρήματα. Με την υποβολή της φόρμας, το ποσό που όρισε ο χρήστης αφαιρείται από τον λογαριασμό του και μεταφέρεται στον λογαριασμό του παραλήπτη. Ο κακόβουλος χάκερ της ιστορίας (δηλαδή εμείς ;) θα πραγματοποιήσει μια επίθεση CSRF για να κλέψει χρήματα από το θύμα, αξιοποιώντας αυτήν ακριβώς τη φόρμα. Η κλοπή θα συντελεστεί αυτόματα, μέσω κάποιων αιτημάτων που θα πραγματοποιήσει το θύμα *χωρίς* να αντιληφθεί το παραμικρό.

Το τραπεζικό website

Όπως αντιλαμβάνεστε, θα σας δώσουμε όλα τα αρχεία που απαιτούνται για τη δημιουργία του τραπεζικού website και της σελίδας του επιτιθέμενου. Για την αξιοποίησή τους, όμως, θα χρειαστεί να κατεβάσετε και να εγκαταστήσετε το πακέτο που αναφέραμε προηγουμένως (WAMP). Αφού ξεμπερδέψετε με την εγκατάσταση και ξεκινήσετε το WAMP, θα εμφανιστεί ένα σχετικό εικονίδιο στο system tray.

Τώρα μπορούμε να ξεκινήσουμε το στήσιμο του τραπεζικού website. Για αρχή, θα κατασκευάσουμε τη βάση δεδομένων με τους πελάτες. Αυτή η δουλειά θα γίνει με τη βοήθεια του εργαλείου phpMyAdmin, που διευκολύνει τη δημιουργία και τη διαχείριση βάσεων της MySQL. Για την εκκίνηση του εργαλείου αρκεί να κάνουμε κλικ στο εικονίδιο του WAMP και να επιλέξουμε το… phpMyAdmin! Για τη σύνδεση στο περιβάλλον του προγράμματος οφείλουμε να δώσουμε σαν username το root και να αφήσουμε το πεδίο του password κενό. Στη συνέχεια, πρέπει να επισκεφτούμε την καρτέλα "Κώδικας SQL" και στο μεγάλο text box που θα δούμε να εισάγουμε τα ακόλουθα:

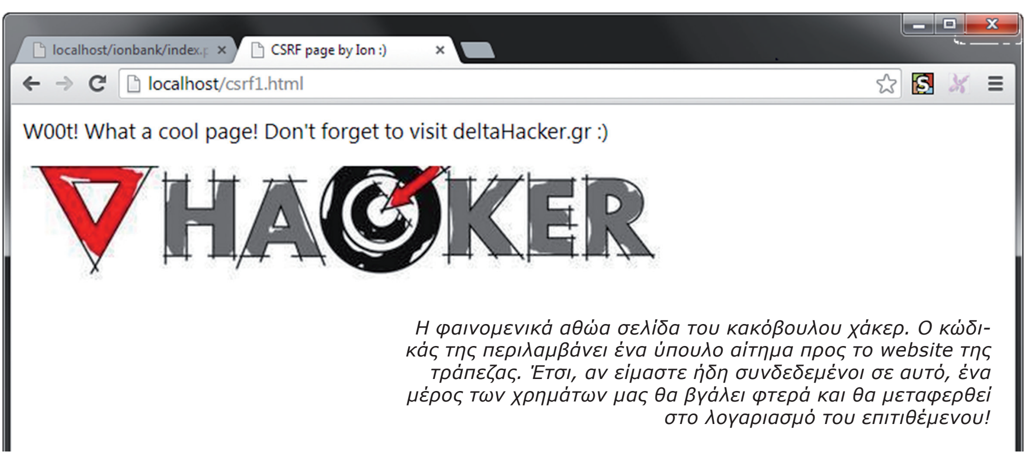
```
CREATE DATABASE `ionbank`;
USE `ionbank`;

CREATE TABLE IF NOT EXISTS `users` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `customer_code` bigint(20) NOT NULL,
  `username` varchar(60) NOT NULL,
  `password` char(40) NOT NULL,
  `amount` float NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY (`customer_code`)
) AUTO_INCREMENT=7;
```

```
INSERT INTO `users` (`id`, `customer_code`, `username`, `password`,
`amount`) VALUES
(1, 111000, 'george', '57b2ad99044d337197c0c39fd3823568ff81e48a',
1200.33),
(2, 111001, 'john', '7c4a8d09ca3762af61e59520943dc26494f8941b', 2330.45),
(3, 111002, 'papadopoulos', 'd72bfe92cf0ec06a4fad6db8d339d6c75ef1ef00',
21220.4),
(4, 111003, 'maria', '51abb9636078defbf888d8457a7c76f85c8f114c', 990.35),
(5, 111004, 'stelios', 'c19d95ea822d83b01cdfb1ce20abcfda77d65d42',
8734.45),
(6, 111999, 'hacker', '4dcc4173d80a2817206e196a38f0dbf7850188ff', 120.5);
```

Αμέσως μετά, αρκεί να πατήσουμε το κουμπί "Εκτέλεση". Ακόμα κι αν δεν έχετε καμία απολύτως ιδέα γύρω από τη γλώσσα SQL, αν μελετήσετε λίγο τον κώδικα θα καταλάβετε αμέσως τι κάνει: Δημιουργεί μια βάση με το όνομα ionbank και μέσα σ' αυτή προσθέτει έναν πίνακα με το όνομα users. Σε αυτόν τον πίνακα καταχωρούνται έξι χρήστες, μαζί με τα στοιχεία που αναφέραμε νωρίτερα (αναγνωριστικό, κωδικός, username, password και χρηματικό ποσό). Όπως βλέπετε, ο τελευταίος χρήστης έχει το (καθόλου εμπνευσμένο) όνομα hacker. Πρόκειται για τον λογαριασμό που έχει ανοίξει ο επιτιθέμενος στο website της τράπεζας και στον οποίο θα μεταφέρονται αργότερα τα κλοπιμαία.

Μετά από τα παραπάνω, η βάση δεδομένων για το website της τράπεζας είναι έτοιμη. Αυτό που χρειαζόμαστε τώρα είναι το ίδιο το website. Μπορείτε να κατεβάσετε τα αρχεία που το απαρτίζουν από το `http://bit.ly/dh027ionbankvuln`. Αποσυμπιέζοντας το συγκεκριμένο πακέτο ZIP θα πάρετε έναν κατάλογο με το όνομα ionbank. Ολόκληρος αυτός ο κατάλογος πρέπει να τοποθετηθεί στο λεγόμενο Document Root του web server. Μετά από αυτή την αντιγραφή, το website της τράπεζας Ionbank θα είναι έτοιμο και πλήρως λειτουργικό! Για να το δείτε αρκεί να ανοίξετε έναν browser και να μεταβείτε στη διεύθυνση `http://localhost/ionbank`. Όπως αντιλαμβάνεστε, η αναλυτική περιγραφή της κατασκευής ενός website ξεφεύγει από το θέμα μας. Ωστόσο, για να διευκολύνουμε τη μελέτη σας, θα πούμε δυο λόγια για τα αρχεία που απαρτίζουν το website της τράπεζας Ionbank.



Η φαινομενικά αθώα σελίδα του κακόβουλου χάκερ. Ο κώδικάς της περιλαμβάνει ένα ύπουλο αίτημα προς το website της τράπεζας. Έτσι, αν είμαστε ήδη συνδεδεμένοι σε αυτό, ένα μέρος των χρημάτων μας θα βγάλει φτερά και θα μεταφερθεί στο λογαριασμό του επιτιθέμενου!

- **config.php.** Περιέχει τα απαραίτητα στοιχεία για τη σύνδεση στη βάση δεδομένων MySQL, όπου βρίσκονται αποθηκευμένα τα δεδομένα των πελατών. Σε αυτό το αρχείο ορίζεται και η μεταβλητή MYSQL_PASS, στην οποία πρέπει να δώσετε τον κωδικό του χρήστη root (της MySQL).
- **functions.php.** Σε αυτό το αρχείο περιέχεται ο κώδικας για διάφορες βοηθητικές συναρτήσεις, που χρησιμοποιούνται από τον υπόλοιπο κώδικα του website. Για παράδειγμα, θα δείτε τη συνάρτηση που τυπώνει τη φόρμα σύνδεσης (login), τη φόρμα μεταφοράς χρημάτων κ.λπ.
- **index.php.** Πρόκειται για την κύρια σελίδα του website. Ο κώδικας είναι ελάχιστος και πολύ απλός, γιατί το website που κατασκευάσαμε εκτελεί ελάχιστες λειτουργίες κι απλές λειτουργίες.
- **login.php.** Σε αυτή τη σελίδα φτάνουν τα δεδομένα που υποβάλλουμε στη φόρμα σύνδεσης. Ο κώδικας ελέγχει αν τα εισερχόμενα δεδομένα αντιστοιχούν σε κάποιον χρήστη της βάσης δεδομένων του website κι αποφασίζει το αν θα επιτραπεί η σύνδεση ή όχι.
- **transfer.php.** Αυτή η σελίδα χειρίζεται τα δεδομένα που παρέχει ο χρήστης στη φόρμα μεταφοράς χρημάτων.
- **logout.php.** Μια εξίσου απλή σελίδα, από όπου πραγματοποιείται η αποσύνδεση του χρήστη από το website.

Πριν προχωρήσουμε οφείλουμε να σταθούμε σε μία σημαντική λεπτομέρεια: Τα δεδομένα που εισάγουμε στη φόρμα μεταφοράς χρημάτων καταλήγουν στη σελίδα transfer.php, με τη μέθοδο GET. Αυτό φαίνεται και στο αρχείο functions.php, όπου κατασκευάζεται η φόρμα. Η γραμμή 33 δηλώνει ότι τα δεδομένα αποστέλλονται στη σελίδα transfer.php κι ότι η μέθοδος μεταφοράς είναι η GET:

```
<form name="transfer_form" method="get" action="transfer.php">
```

Αυτό σημαίνει πως όταν ο χρήστης πατάει το κουμπί υποβολής της φόρμας, τα προς μετάδοση δεδομένα τοποθετούνται στο URL. Έτσι, επειδή η συγκεκριμένη φόρμα ζητάει το χρηματικό ποσό που θα μεταφερθεί και τον αναγνωριστικό κωδικό του παραλήπτη, το URL που σχηματίζεται έχει την ακόλουθη μορφή:

```
http://localhost/ionbank/transfer.php?amount=XXX&recipient=YYY
```

Η σελίδα – παγίδα

Αυτό που χρειαζόμαστε τώρα, για να προχωρήσουμε στην πειραματική μελέτη των επιθέσεων CSRF, είναι η κακόβουλη σελίδα. Μπορείτε να κατεβάσετε μια πρώτη εκδοχή αυτής της σελίδας από το http://bit.ly/dh027csrf1. Στο πακέτο θα βρείτε μόνο μια εικόνα και μια απλή σελίδα HTML. Αυτά τα αρχεία πρέπει να τοποθετηθούν σε έναν νέο κατάλογο εντός του Document Root ή και "χύμα". Ας ρίξουμε μια ματιά στον κώδικα της κακόβουλης σελίδας:

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>CSRF page by Ion :)</title>
</head>
<body>
    <p>W00t! What a cool page! Don't forget to visit deltaHacker.gr :)</p>
```

```
    <a href="http://deltahacker.gr"><img src="deltahacker.jpg"/></a>
    <img src="http://localhost/ionbank/transfer.
php?amount=100&recipient=111999" width="1px" height="1px"/>
</body>
</html>
```

Όλη η βρώμικη δουλειά γίνεται μέσα στη δεύτερη ετικέτα (ή tag αν προτιμάτε) <img>. Το πρώτο tag αυτού του είδους φορτώνει την εικόνα που συνοδεύει τη σελίδα και την τοποθετεί μέσα σε μία ετικέτα <a>. Με αυτόν τον τρόπο δημιουργείται ένας σύνδεσμος προς τη σελίδα του περιοδικού deltaHacker. Μέχρι εδώ καλά. Δεν συμβαίνει τίποτα πονηρό. Όμως, η δεύτερη ετικέτα <img> προσπαθεί να φορτώσει κάτι που δεν είναι εικόνα:

```
src="http://localhost/ionbank/transfer.php?amount=100&recipient=111999"
```

Αντί για το URL ενός αρχείου εικόνας, συναντάμε το URL προς τη σελίδα transfer.php, του website της τράπεζας! Μάλιστα, το URL περιλαμβάνει και τις δύο απαιτούμενες παραμέτρους, ώστε να μεταφερθούν χρήματα στο λογαριασμό με κωδικό "111999". Αναρωτιέστε σε ποιον ανήκει αυτός ο λογαριασμός; Μα, φυσικά, στον πελάτη ονόματι hacker ;)

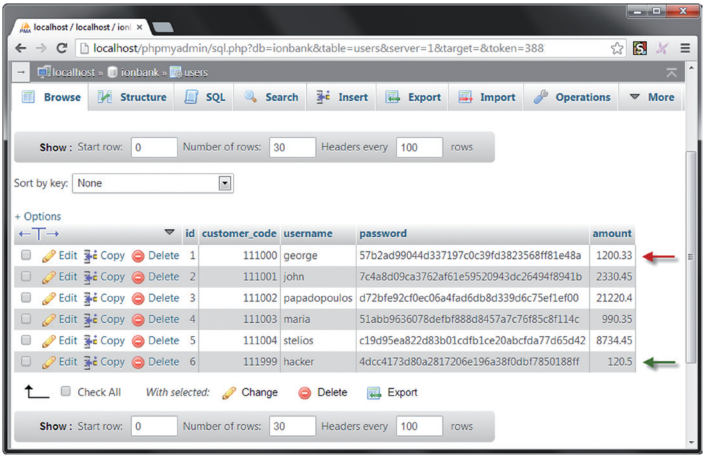
Δοκιμή

Τώρα που στήσαμε όλη την υποδομή του πειράματος, μπορούμε να προχωρήσουμε στις δοκιμές. Αν η σελίδα της τράπεζας δεν τσεκάρει την προέλευση των εισερχόμενων αιτημάτων, οι πελάτες της κινδυνεύουν. Ας υποθέσουμε ότι ο κακόβουλος χάκερ καταφέρνει με κάποιο τρόπο να τραβήξει έναν πελάτη της τράπεζας στη σελίδα του. Αν αυτός ο πελάτης δεν είναι συνδεδεμένος στο website της τράπεζας, δεν θα συμβεί απολύτως τίποτα. Για του λόγου το αληθές, μπορείτε να φορτώσετε τη σελίδα `http://localhost/csrf1.html` και στη συνέχεια να τσεκάρετε τη βάση δεδομένων της τράπεζας, μέσα από το εργαλείο phpMyAdmin. Όπως θα διαπιστώσετε, δεν θα πραγματοποιηθεί καμία αλλαγή στις καταθέσεις των χρηστών. Αν όμως το θύμα είχε συνδεθεί στο website της τράπεζας, η επίσκεψη στην κακόβουλη σελίδα θα του στοίχιζε 100 ευρώ! Για να δείτε την κλοπή στην πράξη, συνδεθείτε στο website της τράπεζας (`http://localhost/ionbank/index.php`) με username το george και password το p@ssw0rd. Μόλις συνδεθείτε, θα δείτε ότι το υπόλοιπο του λογαριασμού του χρήστη George ανέρχεται στα 1200,33 ευρώ. Επισκεφθείτε τώρα την κακόβουλη σελίδα. Κατά την επίσκεψη δεν θα δείτε τίποτα ανησυχητικό στην οθόνη. Η κλεψιά όμως θα έχει πραγματοποιηθεί! Αν επιστρέψετε στο tab με το website της τράπεζας και κάνετε μια ανανέωση, θα δείτε ότι το υπόλοιπο του λογαριασμού έχει μειωθεί στα 1100,33 ευρώ.

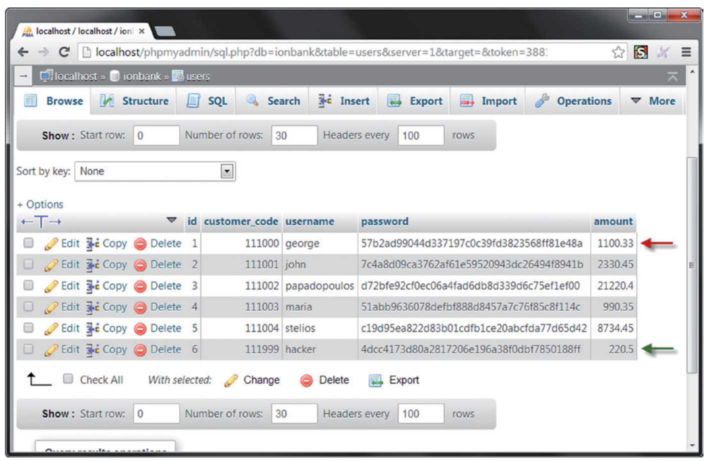
Εναλλακτική επίθεση

Η επίθεση που μελετήσαμε λειτούργησε άψογα. Τι θα γινόταν όμως αν το website της τράπεζας λειτουργούσε λίγο διαφορετικά; Αν η φόρμα μεταφοράς χρημάτων χρησιμοποιούσε τη μέθοδο POST, η κακόβουλη σελίδα που εξετάσαμε δεν θα κατάφερνε απολύτως τίποτα. Αυτό όμως δεν σημαίνει ότι θα ήταν αδύνατη μια επίθεση CSRF! Κατ' αρχάς, αν το website της τράπεζας χρησιμοποιούσε τη μέθοδο POST, η γραμμή 33 του αρχείου functions.php θα ήταν κάπως έτσι:

```
<form name="transfer_form" method="post" action="transfer.php">
```



Παρατηρείστε τα χρηματικά ποσά των πελατών george και hacker. Η περιουσία του hacker αναμένεται να αυξηθεί, μετά από μία αντίστοιχη μείωση της περιουσίας του george!



Μετά την επίσκεψη στην κακόβουλη σελίδα —και χωρίς να εμφανιστεί τίποτα το ανησυχητικό—, ένα μέρος των χρημάτων του George μεταφέρθηκαν στο λογαριασμό του hacker. Η κλοπή επιβεβαιώνεται κι από τα στοιχεία της βάσης δεδομένων της τράπεζας.

Σας προτείνουμε να κάνετε αυτή την αλλαγή, για να μπορέσετε στη συνέχεια να πραγματοποιήσετε μια νέα δοκιμή της επίθεσης. Στην περίπτωση που εξετάζουμε, ο κακόβουλος χάκερ θα έπρεπε να κατασκευάσει μια εντελώς διαφορετική σελίδα. Εσείς, μπορείτε να την κατεβάσετε από το http://bit.ly/dh027csrf2 και να την τοποθετήσετε πάλι στο Document Root του web server. Ας δούμε όμως πώς λειτουργεί αυτή η σελίδα. Όπως έχουμε πει, η μέθοδος POST δεν μεταφέρει τα δεδομένα μέσα στο URL. Επομένως, η λύση δεν βρίσκεται στη σύνθεση ενός νέου URL. Αυτή τη φορά, ο επιτιθέμενος πρέπει να τοποθετήσει στη σελίδα του μια ολόκληρη φόρμα, με τα ίδια ακριβώς στοιχεία που περιλαμβάνει η φόρμα στο website της τράπεζας. Αυτή η φόρμα θα είναι εξ ορισμού συμπληρωμένη με τις κατάλληλες τιμές και φυσικά δεν θα εμφανίζεται ποτέ στον επισκέπτη (θα είναι κρυφή). Προσέξτε τώρα το ακόλουθο απόσπασμα από την ανανεωμένη κακόβουλη σελίδα:

```
<form name="steal_money_form" method="post" action="http://localhost/
ionbank/transfer.php">
```

Πρόκειται για την πρώτη γραμμή της κρυφής φόρμας. Ως μέθοδος αποστολής έχει επιλεχθεί η POST, ενώ κατά την υποβολή της φόρμας τα δεδομένα αποστέλλονται

στη σελίδα transfer.php της τράπεζας! Επομένως, για να πετύχει το σκοπό της η κακόβουλη σελίδα, αρκεί να πατηθεί το κουμπί υποβολής της κρυφής φόρμας. Όπως αντιλαμβάνεστε, ο επιτιθέμενος έχει φροντίσει *και* γι' αυτό. Αν μελετήσετε τον κώδικα της νέας κακόβουλης σελίδας, θα παρατηρήσετε ότι το κυρίως μέρος της σελίδας (body) ξεκινά ως εξής:

```
<body onload="document.forms['steal_money_form'].submit()">
```

Σε αυτή την εκδοχή της κακόβουλης σελίδας, ο επιτιθέμενος χρησιμοποιεί και λίγη JavaScript. Συγκεκριμένα, χρησιμοποιεί το γεγονός (event) onload, ώστε μόλις φορτωθεί η σελίδα να πραγματοποιήσει αυτόματα μια ενέργεια. Η ενέργεια που πραγματοποιείται αυτόματα είναι η υποβολή (submit) της φόρμας steal_money_form. Πρόκειται για την κρυφή φόρμα που περιγράψαμε νωρίτερα και η υποβολή της οποίας πραγματοποιεί την κλοπή!

Αν κάνετε μια δοκιμή με τη νέα κακόβουλη σελίδα, θα διαπιστώσετε ότι η κλοπή γίνεται κανονικά. Αυτή τη φορά, όμως, η ύπουλη δράση της κακόβουλης σελίδας αποκαλύπτεται. Μόλις φορτωθεί και υποβληθεί η κρυφή φόρμα (αυτή για τη μεταφορά των χρημάτων), ο browser θα κατευθυνθεί (redirect) προς το website της τράπεζας. Αυτό είναι σίγουρο ότι θα ανησυχούσε ακόμα και τον πιο καλόπιστο και ανυποψίαστο χρήστη. Ωστόσο, η κλοπή θα είχε πραγματοποιηθεί.

Άμυνα για τα websites

Η τεχνική με την οποία μπορούμε να προστατεύσουμε ένα website από τις επιθέσεις CSRF προβλέπει τη χρήση μιας πρόσθετης παραμέτρου. Μαζί με τα υπόλοιπα στοιχεία της εκάστοτε φόρμας στέλνουμε κι ένα στοιχείο που ονομάζεται nonce (σύντμηση της φράσης "number used only once"). Πρόκειται για ένα αλφαριθμητικό που παράγεται τυχαία και το οποίο αποθηκεύεται στην τρέχουσα συνεδρία (session) του χρήστη με το website. Όταν υποβάλλεται μια φόρμα, ο κώδικας του website περιμένει το στοιχείο nonce και φυσικά ελέγχει αν η τιμή του ταιριάζει με αυτήν που έχει αποθηκευτεί στο session του συνδεμένου χρήστη. Αν είναι όλα καλά, το αίτημα γίνεται αποδεκτό. Σε διαφορετική περίπτωση, ο κώδικας του website θεωρεί ότι το αίτημα προέρχεται από μη αξιόπιστη πηγή και το αγνοεί!

Ας δούμε πώς μπορούμε να υλοποιήσουμε έναν τέτοιο μηχανισμό άμυνας. Θα εργαστούμε στο website της τράπεζας και συγκεκριμένα στο αρχείο functions.php. Κατ' αρχάς, θα χρειαστούμε μία συνάρτηση που θα παράγει το στοιχείο nonce:

```
function addNonce($salt_size)
{
    if(isset($_SESSION['nonce']))
    {
            $nonce = $_SESSION['nonce'];
    }
    else
    {
            $nonce = md5(mcrypt_create_iv($salt_size, MCRYPT_DEV_
URANDOM));
            $_SESSION['nonce'] = $nonce;
```

```
        }
        return $nonce;
    }
```

Για την υλοποίηση αυτή της συνάρτησης χρησιμοποιούμε μια άλλη συνάρτηση (mcrypt_create_iv), η οποία ανήκει στη βιβλιοθήκη mcrypt και παράγει τυχαίους κωδικούς. Ο κωδικός που παράγεται αποθηκεύεται μαζί με τα υπόλοιπα δεδομένα του session. Παρεμπιπτόντως, τα δεδομένα του session φυλάσσονται πάντα σε έναν πίνακα με το όνομα $_SESSION. Έχοντας ετοιμάσει τη συνάρτηση παραγωγής του nonce, πρέπει να εξασφαλίσουμε ότι θα εκτελείται αυτόματα κάθε φορά που κατασκευάζεται η φόρμα μεταφοράς χρημάτων. Αυτό γίνεται εύκολα, προσθέτοντας στον κώδικα της φόρμας την ακόλουθη γραμμή:

```
<input type="hidden" name="nonce" value="'.addNonce(10).'"/>
```

Η παραπάνω γραμμή εισάγει ένα νέο στοιχείο στη φόρμα. Το όνομα του στοιχείου είναι nonce και για την τιμή του καλείται η συνάρτηση adNonce. Μέχρι στιγμής, το ανανεωμένο website της τράπεζας παράγει αυτόματα ένα στοιχείο nonce και το τοποθετεί στη φόρμα μεταφοράς χρημάτων. Επομένως, πρέπει να φτιάξουμε και τον μηχανισμό που θα ελέγχει για την ύπαρξη του nonce. Αυτός ο μηχανισμός θα τοποθετηθεί σε μία νέα συνάρτηση, στο αρχείο functions.php:

```
function checkNonce()
{
    $nonce = getParameter('nonce');

    if(isset($_SESSION['nonce']) && $_SESSION['nonce'] == $nonce)
        return TRUE;

    return FALSE;
}
```

Η παραπάνω συνάρτηση ελέγχει αν η παράμετρος nonce που μεταδόθηκε με τα δεδομένα μιας φόρμας (getParameter('nonce')) ταυτίζεται με την τιμή nonce που έχει αποθηκευτεί στην τρέχουσα συνεδρία ($_SESSION['nonce']). Έχοντας ετοιμάσει κι αυτή τη συνάρτηση, απομένει η αξιοποίησή της. Όπως καταλαβαίνετε, ο έλεγχος για την ύπαρξη αλλά και την τιμή του nonce πρέπει να πραγματοποιείται κάθε φορά που υποβάλλεται η φόρμα μεταφοράς χρημάτων. Επομένως, η παραπάνω συνάρτηση πρέπει να καλείται κάθε φορά που φορτώνεται η σελίδα transfer.php. Έτσι, στον κώδικα αυτής της σελίδας και συγκεκριμένα στη γραμμή όπου γίνεται έλεγχος για το αν έχει συνδεθεί ο χρήστης, θα πρέπει να καλείται και η συνάρτηση checkNonce. Η σχετική γραμμή θα πρέπει να γίνει κάπως έτσι:

```
if(!checkLogin() || !checkNonce())
```

Μετά κι από αυτή την επέμβαση, το τραπεζικό website θα είναι ασφαλές απέναντι στις επιθέσεις CSRF!

Αν θέλετε να αποφύγετε τη χειροκίνητη προσθήκη των παραπάνω βελτιώσεων, μπορείτε να κατεβάσετε την ενισχυμένη/ασφαλή εκδοχή του τραπεζικού website από το http://bit.ly/dh027ionbanksec. Αφού αποσυμπιέσετε το πακέτο, αντιγράψτε τον κατάλογο ionbank στο Document Root, "πάνω" από την παλιά εκδοχή του website.

Τώρα, μπορείτε να συνδεθείτε στο ανανεωμένο website και να δοκιμάσετε την επίθεση με τη σελίδα csrf1.html. Όπως θα διαπιστώσετε, η επίθεση θα αποτύχει και ο λογαριασμός του θύματος θα παραμείνει αναλλοίωτος.

Άμυνα για τους χρήστες

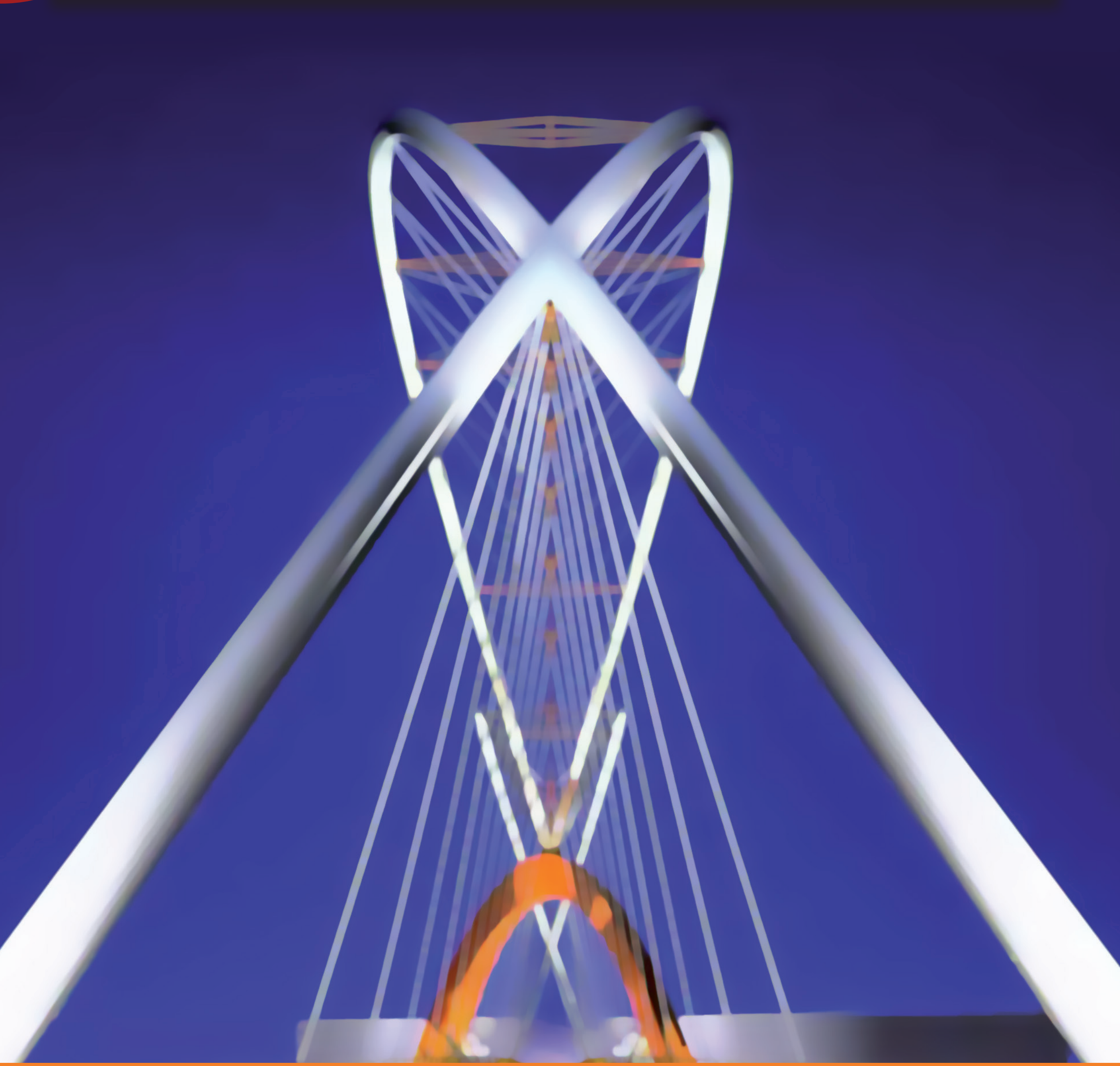
Η επιτυχία ή η αποτυχία μιας επίθεσης CSRF δεν καθορίζεται μόνο από τον κώδικα του εκάστοτε website. Οι απαιτούμενοι μηχανισμοί προστασίας ενδέχεται να υπάρχουν, αλλά μπορεί και να 'ναι ελλιπείς ή ακόμη και να απουσιάζουν παντελώς. Επομένως, μεγάλο μερίδιο ευθύνης έχει κι ο εκάστοτε χρήστης. Ας δούμε μερικές απλές αλλά χρήσιμες συμβουλές.

1. **Κάντε πάντα Logout.** Οι επιθέσεις CSRF απαιτούν τη σύνδεση του θύματος με το ευπαθές website. Μην τους κάνετε τη χάρη! Όταν τελειώνετε την εργασία σας σε κάποιο website, φροντίστε πάντα να κάνετε αποσύνδεση. Αυτό σημαίνει ότι θα πρέπει να ξεχάσετε και το κουμπάκι "να με θυμάσαι" (remember me).
2. **Απόρριψη δεδομένων περιήγησης.** Οι σύγχρονοι browsers προσφέρουν την επιλογή της αυτόματης απόρριψης (της μη αποθήκευσης) των στοιχείων περιήγησης και κατ' επέκταση των cookies. Αυτό το μέτρο βέβαια είναι λίγο ακραίο, καθώς ο browser δεν θα θυμάται ποτέ και τίποτα από όσα κάνατε. Μπορείτε όμως να το εφαρμόσετε μόνο σε έναν από τους εγκατεστημένους browsers του συστήματος. Αργότερα, θα χρησιμοποιείτε τον συγκεκριμένο browser για τη σύνδεση στα κρίσιμα websites (web banking, PayPal κ.λπ.) και κάποιον άλλο για την περιήγηση στα υπόλοιπα sites. Στον Firefox, η ρύθμιση που περιγράφουμε επιτυγχάνεται ακολουθώντας τη διαδρομή Tools → Options → Privacy → History κι επιλέγοντας "Never remember history". Στον Chrome μπορείτε να χρησιμοποιείτε το λεγόμενο Incognito mode. Πατήστε το κουμπί του μενού πάνω δεξιά και επιλέξτε "New incognito window". Προσοχή: Αυτό το μέτρο λειτουργεί βοηθητικά και δεν καθιστά την περιήγηση απόλυτα ασφαλή.
3. **Ενημερωθείτε για το phishing.** Κανένας δεν γνωρίζει όλα τα κενά ασφάλειας που υπάρχουν ή ενδέχεται να προκύψουν στους browsers και στα διάφορα websites. Επομένως, είναι πολύ σημαντικό να αντιλαμβάνεστε έγκαιρα τις πιθανές προσπάθειες εξαπάτησης. Για το σκοπό αυτό θα σας προτείναμε να διαβάσετε τα άρθρα "Πώς να μην πιαστείτε στον... ξύπνιο" (τεύχος 018) και "Ώρα για ψάρεμα" (τεύχος 014). Επίσης, κάνετε πότε πότε και μια βόλτα από το `http://securitycartoon.com`, όπου υπάρχει ένα ωραίο κόμικ με θέμα το phishing :D

Επίλογος...

Οι επιθέσεις CSRF στηρίζονται σε μια απλή και σχετικά εύκολα υλοποιήσιμη ιδέα. Όπως είδατε, καταφέραμε να πραγματοποιήσουμε επιθέσεις χρησιμοποιώντας τις μεθόδους GET και POST, χωρίς να αντιμετωπίσουμε κάποιο ιδιαίτερο πρόβλημα. Μπορείτε να μελετήσετε τον κώδικα που γράψαμε, αλλά θα σας προτείναμε να σκεφτείτε το πρόβλημα και μόνοι σας, όπως επίσης και να δοκιμάσετε τις δικές σας ιδέες για την αντιμετώπισή του. Εμείς πάντως θα χαρούμε πολύ να ακούσουμε τις σκέψεις σας, στα γνωστά στέκια επικοινωνίας του περιοδικού. Μέχρι την επόμενη διαδικτυακή μας περιπέτεια... stay secure!

Skill: Beginner
Tags: hub, switch, bridge, learning bridge, router, wireless access point, firewall, modem, OSI model, physical layer, data link layer, network layer

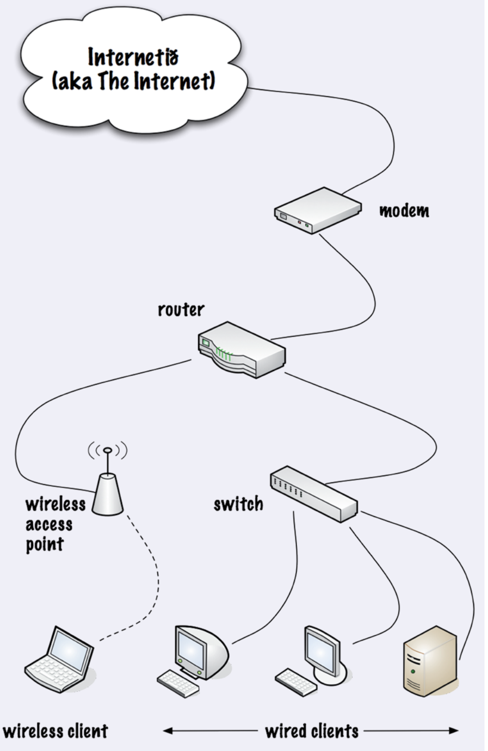


Οι δικτυακές γέφυρες, αλλά από την αρχή

Στον κόσμο των δικτύων υπάρχουν κάποιες τεχνολογίες για τις οποίες ναι μεν μπορεί κάποτε να έχεις ακούσει κάτι, ωστόσο ποτέ δεν τους έχεις δώσει σημασία για πάνω από 8 δευτερόλεπτα. Κι όμως, αν τύχει και μάθεις καλύτερα σε τι αφορούν, τότε συναρπάζεσαι από τις πιθανές εφαρμογές και η άποψή σου για το τι είναι, π.χ., τοπικό δίκτυο, αλλάζει για πάντα. Νομίζετε ότι υπερβάλλουμε; Ελάτε μαζί μας σε μια ξενάγηση σ' ένα τοπίο όπου επικρατούν οι γέφυρες — και μετά τα ξαναλέμε :)

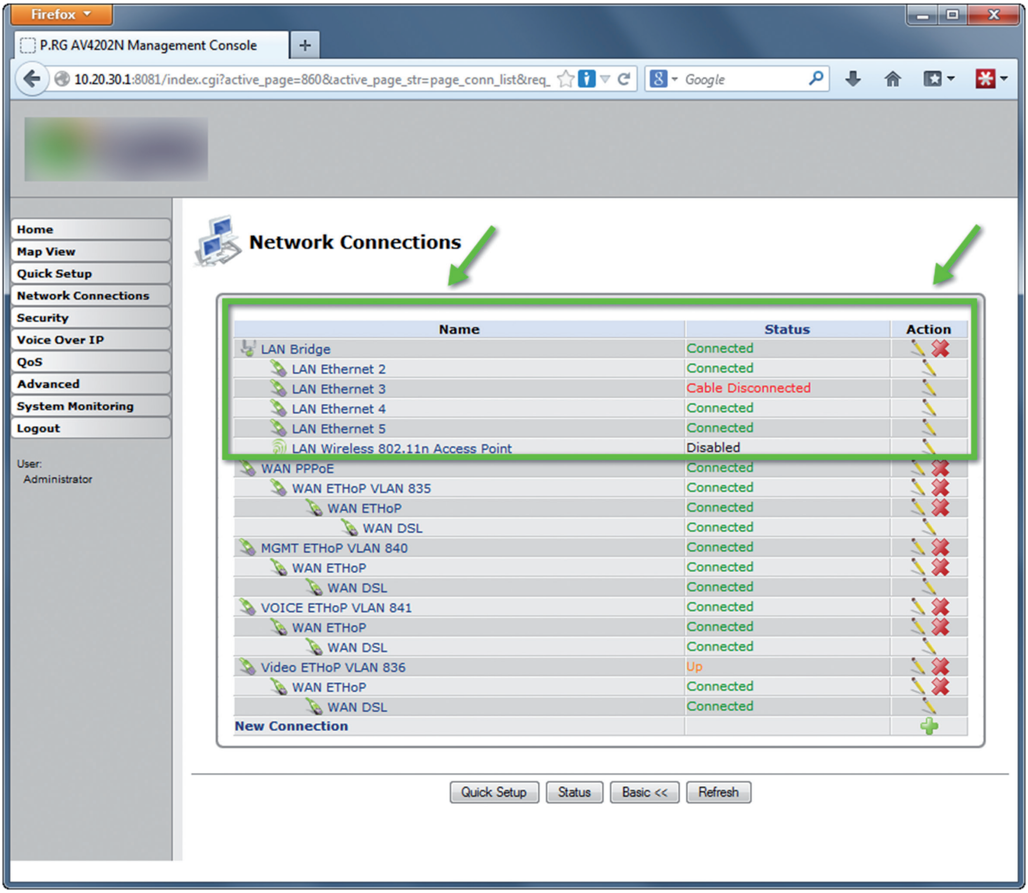
by subZraw

Το modem που έχετε στο σπίτι σας κατά πάσα πιθανότητα είναι *και* δρομολογητής (router). Είναι μάλιστα τόσο συνηθισμένα τα επονομαζόμενα modem/routers, που συχνά μιλάμε μόνο για modem ή μόνο για router και στην πραγματικότητα εννοούμε μια συσκευή που ενσωματώνει τις λειτουργίες και των δύο. Μερικά χρόνια πριν, πάντως, δεν ήταν καθόλου παράξενο να βάζει κάποιος νέα σύνδεση Ίντερνετ, κι από τον ISP να παίρνει ένα σκέτο (DSL) modem. Αν ο ίδιος "κάποιος" ήθελε Ίντερνετ για έναν υπολογιστή και μόνο, τότε λίγο —αν όχι καθόλου— ενδιαφερόταν για το ότι διαθέτει ένα απλό μοντεμάκι, χωρίς δυνατότητες routing (και χωρίς firewall). Αν όμως κάποια στιγμή επιθυμούσε να μοιράσει το Ίντερνετ σε περισσότερα από ένα μηχανήματα, τότε είτε αγόραζε έναν ξεχωριστό router —ενδεχομένως και με δυνατότητες wireless—, είτε μετέτρεπε έναν ολόκληρο υπολογιστή σε router. Σε κάθε περίπτωση, η ιδέα είναι ότι στην πρώτη γραμμή έχουμε το modem, το οποίο βλέπει απευθείας το Ίντερνετ, πίσω από το modem έχουμε τον router, που είναι αυτόνομη συσκευή ή υπολογιστής, ενώ πίσω από τον router έχουμε έναν ή περισσότερους υπολογιστές ή συσκευές, οι οποίοι χάρη στον router μπορούν να επικοινωνούν μεταξύ τους αλλά και να βγαίνουν στο Ίντερνετ μοιραζόμενοι, ουσιαστικά, μία και μόνο σύνδεση (broadband).



Σχήμα 1
Στο δρόμο προς το Internet, οι υπολογιστές ενός τοπικού δικτύου συναντούν ένα wireless access point (WAP) κι ένα switch, μετά καταλήγουν όλοι στον router, εκείνος μιλάει με το modem κι αυτό, με τη σειρά του, διατηρεί τη σύνδεση στο Internet. Στην πλειονότητα των οικιακών τοπικών δικτύων, το WAP κατά πάσα πιθανότητα ενσωματώνεται από τον router, ενώ στη θέση του switch έχουμε μια γέφυρα η οποία, υπό μία έννοια, ομαδοποιεί το WAP με τα Ethernet ports του router. Επίσης, ο router μάλλον ενσωματώνει και το modem. Αν τώρα σκοπεύετε να γυρίσετε την πλάτη στη συσκευή "όλα σ' ένα" που σας έδωσε ο ISP σας και να φτιάξετε τον δικό σας υπέρ-router, τότε κατά πάσα πιθανότητα θα στήσετε κι ένα bridge. Λεπτομέρειες επ' αυτού στο άρθρο που αρχίζει από τη σελίδα 40.

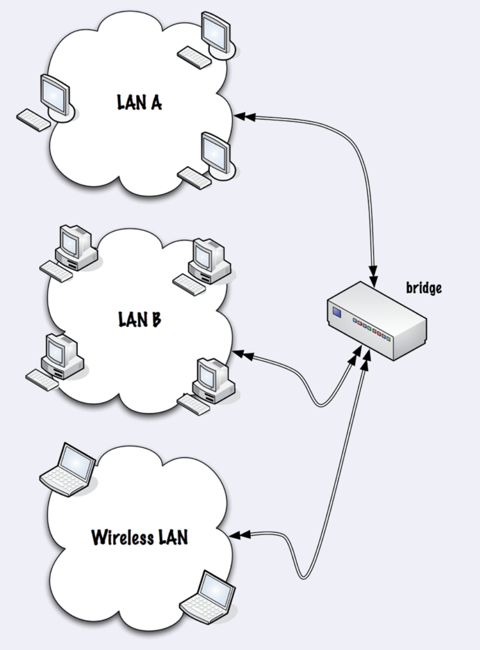
> ***Σημείωση.*** *Η έως τώρα εισαγωγή μάλλον φαντάζει περιττή —αν όχι παράξενη—, για ένα άρθρο με θέμα το bridging. Κρίνεται όμως από τον γράφοντα ως απαραίτητη για μια ομαλή προσέγγιση στο όλο θέμα, ειδικά από τη στιγμή που το περιοδικό το διαβάζουν και φίλοι χωρίς τις γνώσεις που για άλλους θεωρούνται δεδομένες. Προς τους φίλους που τώρα μαθαίνουν, λοιπόν, συστήνουμε υπομονή, επιμονή και φυσικά διαρκή πειραματισμό. Τους υποσχόμαστε ότι, πολύ πιο σύντομα απ' όσο φαντάζονται, θα είναι σε θέση να καταλαβαίνουν πολλά περισσότερα απ' όσα καταλαβαίνουν αυτή τη στιγμή. Στους υπόλοιπους φίλους, συστήνουμε απλά υπομονή :P*



Εικόνα 1
Το bridge που υλοποιεί το modem/router που έχουμε από τον ISP μας, όπως φαίνεται από το web panel της συσκευής. Όπως βλέπετε, γεφυρώνει τέσσερα Ethernet ports και το wireless access point.

Σήμερα, πάντως, είναι σχεδόν βέβαιο ότι όλοι διαθέτουμε modem που ταυτόχρονα είναι *και* router. Πιθανώς μάλιστα ενσωματώνει κι ασύρματο access point (wireless access point ή WAP), ενώ πέρα από τη θύρα WAN, πάνω στην οποία έχουμε το καλώδιο της ιντερνετικής σύνδεσης, υπάρχουν μία ή και περισσότερες θύρες LAN για τους ενσύρματους πελάτες (clients). Αν τώρα το modem/router έχει μία μόνο θύρα LAN, αλλά εμείς θέλουμε να συνδέσουμε περισσότερους από έναν υπολογιστές ενσύρματα, τότε στην εν λόγω θύρα βάζουμε ένα switch. Ένα τυπικό switch διαθέτει τέσσερις θύρες, άρα αμέσως αμέσως πάνω στο modem/router μπορούμε να έχουμε έως και τέσσερις ενσύρματα συνδεδεμένους υπολογιστές — και μην ξεχνάτε και τις συσκευές που συνδέονται ασύρματα.

Πάρτε τώρα ένα modem/router που δεν έχει WAP αλλά διαθέτει περισσότερες από μία θύρες LAN (συγκεκριμένα, πρόκειται για Ethernet ports). Ακόμη και σ' αυτό μπορούμε να προσθέσουμε ένα ή περισσότερα switches, καταλαβαίνουμε όμως ότι για αρκετές περιπτώσεις/σενάρια δεν είναι απαραίτητο. Ας αφήσουμε λοιπόν τα switches



Σχήμα 2
Σχηματική αναπαράσταση μιας συσκευής bridge, που γεφυρώνει τρία διαφορετικά τοπικά δίκτυα. Τα bridges είναι πιο εξειδικευμένα από τα routers και λειτουργούν στα δύο πρώτα επίπεδα του μοντέλου OSI (physical layer και data link layer), ακριβώς κάτω από το επίπεδο όπου λαμβάνει χώρα το routing (network layer).

στην άκρη κι ας αναλογιστούμε το εξής: Η συμπεριφορά όλων των Ethernet ports του modem/router, παραπέμπει στη συμπεριφορά ενός switch. Είναι μάλιστα λογικό να υποθέσουμε ότι η συσκευή έχει ενσωματωμένο κάποιο switch, ωστόσο αυτό δεν ισχύει: Μπορεί όλες μαζί οι θύρες πράγματι να συμπεριφέρονται ως ένα switch, στην πραγματικότητα όμως είναι *μόνο* λογικά γεφυρωμένες. Με τον όρο "λογικά", εννοούμε ότι δεν υπάρχει κάποιο έξτρα hardware που τις γεφυρώνει: Το λειτουργικό σύστημα του modem/router τρέχει κατάλληλο λογισμικό, το οποίο αναλαμβάνει να κάνει το εν λόγω γεφύρωμα. Αν και, διαισθητικά τουλάχιστον, καταλαβαίνουμε τι περίπου είναι αυτό το γεφύρωμα (bridging) που ξαφνικά βάλαμε στην κουβέντα μας, προφανώς δεν έχουμε ακόμα εξηγήσει επαρκώς περί τίνος πρόκειται. Σε λίγο θα το κάνουμε, όμως προς το παρόν λάβετε υπόψη ότι bridging δεν γίνεται μόνο μεταξύ network interfaces[1] του ίδιου τύπου (π.χ., Ethernet interfaces). Πράγματι, κάλλιστα μπορούμε να έχουμε γεφυρωμένα κάποια Ethernet interfaces μαζί με ένα ή περισσότερα wireless interfaces. Στην πραγματικότητα, αυτό ακριβώς συμβαίνει σε κάθε (modem) router που είναι wireless access point κι έχει τουλάχιστον ένα Ethernet port: Όλα τα network interfaces για τους πελάτες, είναι πράγματι γεφυρωμένα. Το γεγονός αυτό ίσως φαίνεται κι από το web panel του router σας (βλ. εικόνα 1).

Σε κάθε περίπτωση, όλοι οι υπολογιστές ή/και οι συσκευές που συνδέονται ενσύρματα ή ασύρματα πάνω στον ίδιο router, έχουν διευθύνσεις IP της αυτής μορφής, δηλαδή διευθύνσεις από το ίδιο subnet (π.χ., 192.168.1.x ή 10.0.0.x κ.ο.κ.). Τις διευθύνσεις IP είναι πολύ πιθανό να τις μοιράζει αυτόματα ο λεγόμενος DHCP server της συσκευής. Όπως και να 'χει, η δρομολόγηση ή/και το μοίρασμα διευθύνσεων γίνονται από ένα κοινό, εικονικό network interface: Είναι αυτό του λεγόμενου bridge interface, το οποίο "ενώνει" όλα τα άλλα network interfaces, ασχέτως αν είναι wireless ή wired. Και μια τελευταία παρατήρηση: Δεν είναι υποχρεωτικό ότι, σε κάθε router, όλα τα network interfaces για τους clients είναι γεφυρωμένα μεταξύ τους. Μπορεί, π.χ., να στήσετε τον δικό σας router με το pfSense ή με το OpenBSD (εντελώς τυχαία παραδείγματα :D) και να μη γεφυρώσετε τίποτε ή να φτιάξετε παραπάνω από μία γέφυρες (bridge groups, όπως ονομάζονται). Πάντως σε έναν απλό, τυπικό (οικιακό) router, θα υπάρχει ένα και μόνο bridge.

1. Ο όρος "interface", όπως τον χρησιμοποιούμε στο παρόν κείμενο, αφορά στο πώς βλέπει το software το hardware. Για παράδειγμα, οι εφαρμογές επικοινωνούν με μια κάρτα Ethernet μέσω του αντίστοιχου Ethernet interface, με μια κάρτα ασύρματης δικτύωσης μέσω του αντίστοιχου wireless interface κ.ο.κ.

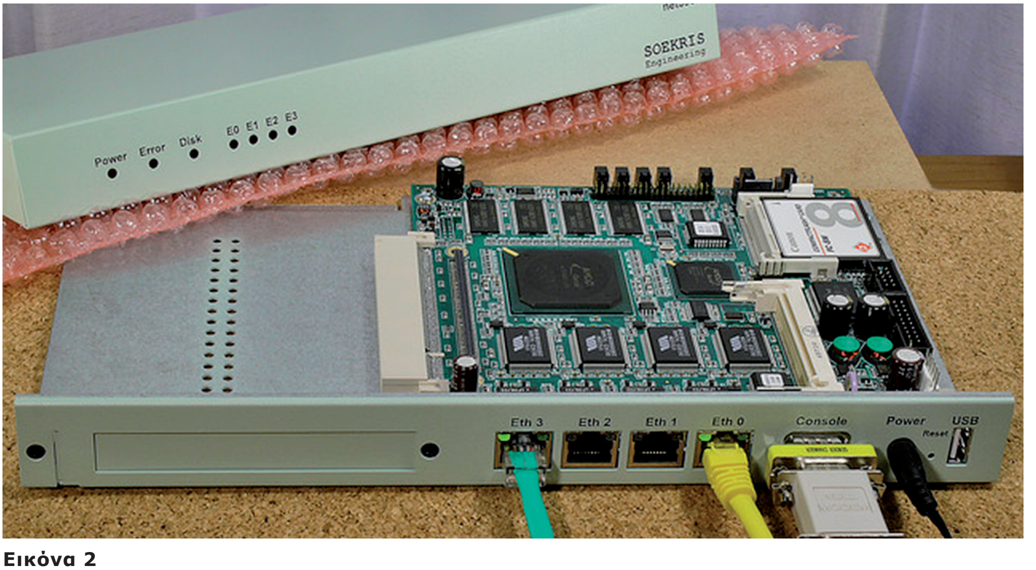
Το bridging, λίγο πιο κοντά

Γενικά, μπορούμε να πούμε ότι το **bridging** ορίζεται ως η τεχνική που επιτρέπει σε δύο ή περισσότερα τοπικά δίκτυα (LANs) να συγκροτούν ένα ευρύτερο, ενιαίο δίκτυο. Παρατηρήστε ότι το **routing** είναι κάτι γενικότερο, αφού αυτό επιτρέπει σε δύο ή περισσότερα δίκτυα να επικοινωνούν μεταξύ τους παραμένοντας, όμως, εντελώς ανεξάρτητα. Πράγματι, το bridging λαμβάνει χώρα στα δύο πρώτα επίπεδα του μοντέλου OSI (physical layer και data link layer), ακριβώς *κάτω* από το επίπεδο όπου λαμβάνει χώρα το routing (network layer). Το bridging επιτυγχάνεται από hardware ή από software. Τα πρακτικά παραδείγματα που δείχνουμε στο άρθρο της σελίδας 40, αφορούν αποκλεστικά σε υλοποιήσεις software.

Στο σημείο αυτό, λογικό είναι ν' αναρωτηθεί κάποιος το εξής: Αντί να έχουμε ένα bridge μεταξύ δύο ή περισσοτέρων δικτύων, γιατί να να μη φτιάξουμε από την αρχή ένα μεγαλύτερο δίκτυο, που να περιλαμβάνει τα επί μέρους; Υπάρχουν διάφοροι λόγοι — και παραθέτουμε τρεις από τους κυριότερους.

Αξιοπιστία. Μια οποιαδήποτε βλάβη σ' ένα (σχετικά) μεγάλο τοπικό δίκτυο, θα έχει ως αποτέλεσμα την αδυναμία επικοινωνίας μεταξύ όλων των μελών του. Αν, από την άλλη, έχουμε δύο ή περισσότερα γεφυρωμένα δίκτυα, τότε πιθανή βλάβη σε κάποιο δίκτυο δεν θα επηρεάσει αρνητικά τη δυνατότητα επικοινωνίας μεταξύ των μελών άλλων δικτύων.

Επιδόσεις. Όσο μεγαλώνει ένα δίκτυο, μ' άλλα λόγια όσα περισσότερα μηχανήματα και συσκευές συμμετέχουν σ' αυτό, τόσο περισότερο μειώνονται οι επιδόσεις του. Μια καλή πρακτική είναι να οργανώνουμε μηχανήματα και συσκευές σε ξεχωριστά δίκτυα, ώστε η ενδοδικτυακή κίνηση πακέτων να είναι σχεδόν πάντα μεγαλύτερη



Εικόνα 2
Εγκαταστήσατε το λειτουργικό της αρεσκείας σας στον ωραιότατο —αλλά και πανίσχυρο— router που φτιάχνετε, ολοκληρώσατε και τις βασικές ρυθμίσεις. Μένει να στήσετε και το bridging, ώστε σε οποιαδήποτε από τα διαθέσιμα Ethernet ports κι αν συνδέετε κάποιον client, αυτός να παίρνει IP από το κοινό pool που διαχειρίζεται ο DHCP server του router. Φυσικά, όλοι οι clients θα μπορούν να επικοινωνούν μεταξύ τους και να βγαίνουν στο Internet — εκτός βέβαια κι αν προσθέσετε κάποιον περιοριστικό κανόνα στο firewall.

από τη διαδικτυακή[2]. Βεβαίως, ακριβώς επειδή σε πάμπολλες περιπτώσεις είναι επιθυμητή *και* αυτή η διαδικτυακή επικοινωνία, τα επιμέρους δίκτυα γεφυρώνονται.

Ασφάλεια. Αναλόγως του περιβάλλοντος, εντός του ίδιου οργανισμού είναι πολύ καλή ιδέα να 'χουμε ξεχωριστά δίκτυα για χρήστες που κάνουν εντελώς διαφορετικές εργασίες ή/και έχουν εντελώς διαφορετικούς ρόλους. Σε μια εταιρεία, π.χ., είναι πιθανό να συναντήσουμε ένα μικρό (ενσύρματο ή/και ασύρματο) δίκτυο για τους επισκέπτες, άλλο ένα για τη γραματεία, άλλο για όσους εμπλέκονται με τον στρατηγικό σχεδιασμό (aka "τα αφεντικά" ή "οι από πάνω"), ξεχωριστό δίκτυο για το marketing, άλλο για το λογιστήριο κ.ο.κ. Ορισμένα ή και όλα από αυτά τα διαφορετικά δίκτυα είναι πιθανό να επικοινωνούν μεταξύ τους, έστω και υπό περιορισμούς. (Στο παιχνίδι, βλέπετε, μπαίνουν και τα firewalls.)

Πώς δουλεύει το bridging;

Ο ρόλος ενός bridge interface, ασχέτως αν υλοποιείται από hardware ή software, είναι να γεφυρώνει ιδεατά δύο ή περισσότερα network interfaces, καθένα από τα οποία ορίζει ένα διαφορετικό δίκτυο. Το bridge πετυχαίνει αυτή τη λογική σύνδεση προωθώντας επιλεκτικά κάποια frames από ένα οποιοδήποτε δίκτυο στα άλλα δίκτυα (του ίδιου bridge group). Δικαιούμαστε να φανταζόμαστε ότι τα bridges συγγενεύουν με τα hubs, αρκεί να έχουμε κατά νου ότι τα bridges είναι πολύ πιο έξυπνα. Πράγματι, ένα δημοφιλές είδος bridge είναι το λεγόμενο learning bridge, το οποίο παρακολουθεί διαρκώς τα headers των εισερχόμενων frames κι έτσι μαθαίνει, σιγά σιγά, ποιοι υπολογιστές βρίσκονται σε καθένα από τα δίκτυα που γεφυρώνει.

Παρακολουθήστε λίγο πώς αντιδρά ένα learning bridge, όταν βλέπει ένα εισεργόμενο frame. Το πρώτο πράγμα που κάνει είναι να ψιλο-εκνευριστεί, αφού ησυχία δεν βρίσκει και κάθε λίγο και λιγάκι βρίσκεται κάποιος που το "τρέχει". Πιστό ωστόσο στο καθήκον του, ξεφυσά και σημειώνει το δίκτυο από το οποίο προέρχεται το frame, καθώς και τη διεύθυνση του υπολογιστή ο οποίος το έστειλε. Αν τώρα το bridge δεν έχει ιδέα για το πού είναι το δίκτυο μέσα στο οποίο βρίσκεται ο υπολογιστής για τον οποίο προορίζεται το frame, λέει κάτι σαν "ουφ, αμάν πια μ' αυτή την αβεβαιότητα" και προωθεί το frame σ' όλα ανεξαιρέτως τα δίκτυα που γεφυρώνει. Αν, πάλι, κοιτάζοντας τις σημειώσεις του, δει ότι το frame είναι να πάει σε συγκεκριμένο δίκτυο, διαφορετικό απ' αυτό που ξεκίνησε, τότε χαρούμενο που δεν θα προκαλέσει περιττή αύξηση της κυκλοφορίας στέλνει το frame απευθείας προς τα εκεί που πρέπει. Αν, τέλος, το frame δεν είναι για να φύγει από το δίκτυο από το οποίο ξεκίνησε το ταξίδι του, τότε το bridge θα χαμογελάσει σκανδαλιάρικα και θα το απορρίψει. "Ααααα, δεν ξέρω, το μηχάνημα για το οποίο προορίζεται το frame είχε κάθε ευκαιρία να το δει και να το πάρει και να το βράσει! Αν τώρα κοιμόταν ή δεν-ξέρω-κι-εγώ-τι, δικό του πρόβλημα", λέει, κι ελπίζει στα επόμενα δέκα χιλιοστά του δευτερολέπτου να μπορέσει να πάρει έναν υπνάκο που τόσο πολύ θα ήθελε.

Με ή χωρίς τις μεταφορές σε καθημερινές καταστάσεις, νομίζουμε ότι η λειτουργία αλλά και η αποστολή των bridges έχουν πλέον γίνει κατανοητές.

Σημειώστε και κάτι τελευταίο, που αφορά στην ασφάλεια. Ορισμένες υλοποιήσεις bridges, πριν προωθήσουν ένα frame κοιτάζουν πρώτα αν περιλαμβάνει ένα IPv4 ή IPv6 datagram. Αν πράγματι περιλαμβάνει, τότε το πακέτο υπόκειται και στους ελέγχους του όποιου firewall. Χάρη στα bridges και στα firewalls, λοιπόν, είναι δυ-

2. Παρατηρήστε ότι δεν χρησιμοποιήσαμε το κεφαλαίο δέλτα, μιας και δεν αναφερόμαστε στην επικοινωνία από και προς το Διαδίκτυο (Internet) αλλά στην επικοινωνία μεταξύ διαφορετικών, τοπικών δικτύων.

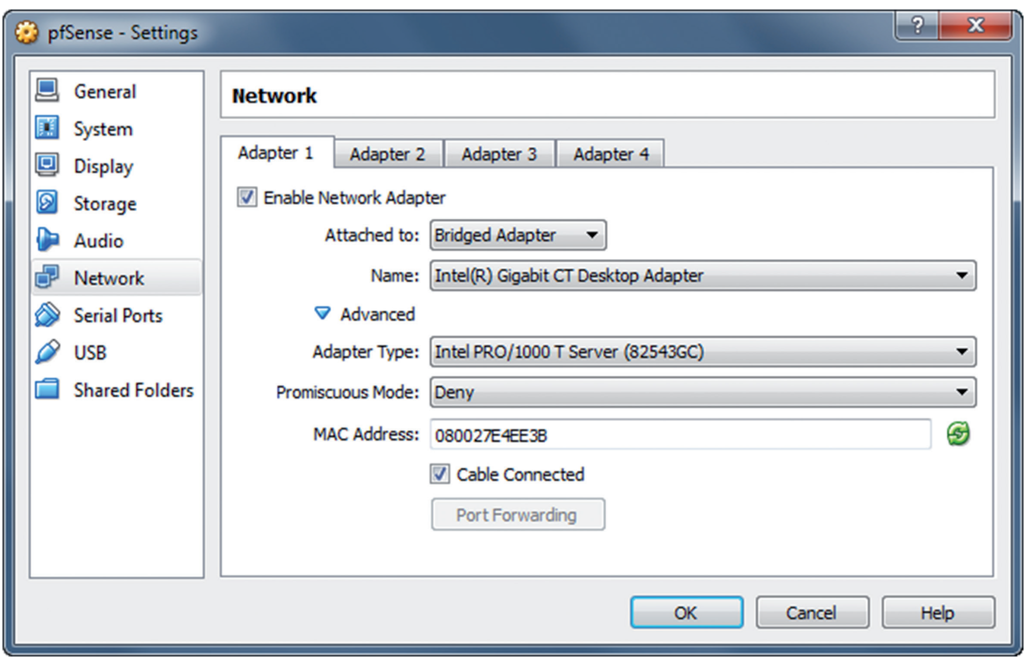
νατόν να φτιάχνουμε περίπλοκους κανόνες ώστε, π.χ., μόνο ένα μηχάνημα από το λογιστήριο να μπορεί να φτάνει στα μηχανήματα του τμήματος μάρκετινγκ κ.ο.κ.

Στην πράξη

Πηγαίνετε τώρα στο άρθρο που αρχίζει από τη σελίδα 40, ώστε να δείτε πώς υλοποιούμε bridges κάτω από δύο διαφορετικά λειτουργικά συστήματα, το pfSense και το OpenBSD, τα οποία αμφότερα είναι άριστα για τη δημιουργία πανίσχυρων κι εξαιρετικά ευέλικτων routers/firewalls.

Υπάρχει εξάλλου και μια χρήση του bridging που αποδεικνύεται ιδιαίτερα βολική σε όσους ασχολούνται με το penetration testing ή έστω σε εκείνους που παίζουν κι εξερευνούν, αν μη τι άλλο για ν' αποδεικνύουν σε γνωστούς και φίλους ότι δεν είναι τόσο ασφαλείς όσο νομίζουν. Πολλά περισσότερα επ' αυτού στο άρθρο που ξεκινά από τη σελίδα 64 του παρόντος τεύχους, στο οποίο συζητάμε για το λεγόμενο wireless bridging και βλέπουμε, αναλυτικά και βήμα προς βήμα, πόσο εύκολες καθιστά τις επιθέσεις Man-in-The-Middle. Πραγματοποιούμε βεβαίως και σχετική επίδειξη, μέσα από το Kali Linux.

Καλή σας διασκέδαση!



Εικόνα 3
Το bridging κατά πάσα πιθανότητα το έχετε δει εν δράσει στον hypervizor που χρησιμοποιείτε — ακόμη κι αν δεν έχετε δώσει τη δέουσα προσοχή. Στο screenshot βλέπουμε τις ρυθμίσεις δικτύωσης ενός VirtualBox VM. Ο Adapter 1 είναι τύπου "Bridged", γεγονός που σημαίνει ότι με το που ενεργοποιείται η εικονική μηχανή το VirtualBox γεφυρώνει αυτόματα τον εικονικό με κάποιον από τους φυσικούς network adapters του host computer. Πρακτικό αποτέλεσμα: Το VM θα ανήκει στο ίδιο τοπικό δίκτυο που ανήκει ο host και θα βλέπει τον φυσικό router που επίσης βλέπει ο host.


Τα επεισόδια του deltaCast, μοιάζουν με ταινίες.
Στη διάρκεια -- όχι στην ποιότητα.
http://deltahacker.gr/category/deltacast

Skill: Beginner
Tags: photocell, RGB LED, normalization

Αναγνώριση χρωμάτων με το Arduino

by GiannouG

Μήπως θέλετε να κατασκευάσετε ένα απλό ρομποτάκι που κινείται ακολουθώντας μια γραμμή στο δάπεδο; Μήπως κάποιο εξελιγμένο, που ακολουθεί ένα συγκεκριμένο χρώμα; Μήπως πάλι θέλετε μόνο έναν απλό τρόπο για να διαβάζετε τα χρώματα των φυσικών αντικειμένων, ώστε να τα αναπαράγετε στον υπολογιστή; Σε κάθε περίπτωση, χρειάζεστε έναν απλό αισθητήρα χρωμάτων. Κάτι τέτοιο χρειαστήκαμε κι εμείς — κι αποφασίσαμε να το κατασκευάσουμε.

Κατά καιρούς έχουμε δει πολλές κατασκευές με το Arduino κι έχουμε επιστρατεύσει διάφορους αισθητήρες. Βέβαια, τα εξαρτήματα που χρησιμοποιούσαμε ήταν σχεδιασμένα για μια απλή λειτουργία, ενώ μπορούσαν να συνδεθούν και να επικοινωνήσουν με το Arduino χωρίς κόπο. Οι αισθητήρες αυτοί, ανάλογα με τη λειτουργία αλλά και την εσωτερική πολυπλοκότητά τους, στοίχιζαν από ελάχιστα μέχρι και αρκετές δεκάδες ευρώ. Σε μια κατασκευαστική περιπέτεια —από εκείνες στις οποίες ο γράφων μπλέκει πολύ συχνά— χρειάστηκε να επιστρατεύσουμε έναν ανιχνευτή χρώματος. Μόνο που το κόστος για ένα τέτοιο εξάρτημα ήταν αρκετά υψηλό, ενώ το πακέτο θα αργούσε να φτάσει. Κάπως έτσι, αναγκαστήκαμε να ψάξουμε για μια εναλλακτική λύση και η αλήθεια είναι ότι την βρήκαμε αμέσως. Για να μην τα πολυλογούμε, αποφασίσαμε να φτιάξουμε μόνοι μας έναν αισθητήρα για τον εντοπισμό/αναγνώριση των χρωμάτων, ξοδεύοντας ελάχιστα ευρώ. Η υλοποίηση που θα παρουσιάσουμε θα βασιστεί σε υλικά που, λίγο πολύ, υπάρχουν στο συρτάρι όλων εκείνων που ασχολούνται με τα ηλεκτρονικά. Σε αυτό το άρθρο, λοιπόν, θα δούμε την κατασκευή του αισθητήρα και θα πούμε δυο λόγια και για τη Φυσική που κρύβεται πίσω από την λειτουργία του. Πριν ξεκινήσουμε, όμως, νομίζουμε ότι πρέπει να μιλήσουμε λίγο παραπάνω για το τι ακριβώς σκοπεύουμε να κάνουμε — και για το πώς θα το πετύχουμε.

Λίγη θεωρία

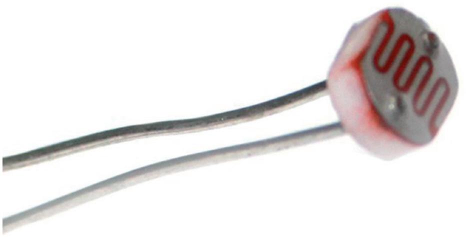
Χρειαζόμαστε μια συσκευή η οποία, με κάποιον τρόπο, θα αναγνωρίζει το χρώμα του αντικειμένου που βρίσκεται μπροστά της. Δεν ζητάμε μεγάλη ακρίβεια στον προσδιορισμό του χρώματος, αλλά μόνο μια εκτίμηση και, κυρίως, την ικανότητα διάκρισης μεταξύ διαφορετικών χωμάτων. Βλέπετε, δεν σκοπεύουμε να τη χρησιμοποιήσουμε για να ρυθμίσουμε κάποιο monitor[1], αλλά για να δώσουμε τη δυνατότητα στο Arduino να ξεχωρίζει το "κόκκινο" απ' το "πράσινο"[2]. Μία λύση θα ήταν να ξοδέψουμε 10 με 15 ευρώ, να περιμένουμε γύρω στον ένα μήνα και να κάνουμε την δουλειά μας ευκολότερα — αλλά όχι με μεγαλύτερη ακρίβεια! Νομίζουμε όμως ότι μας ξέρετε αρκετά καλά: Κάποιες φορές επιλέγουμε τον εύκολο δρόμο και κάποιες άλλες τον δύσκολο. Κριτήριο δεν είναι η ψευτομαγκιά αλλά η πρακτικότητα, το κόστος — και φυσικά η "ποιότητα" του τελικού αποτελέσματος. Επομένως, αν πρόκειται να ξοδέψουμε λεφτά και να πετύχουμε το αποτέλεσμα που θα πετυχαίναμε με λίγη παραπάνω δουλειά από μέρους μας, θα επιλέξουμε το δεύτερο.



Θα μπορούσαμε να αγοράσουμε έναν αισθητήρα TCS3200 και να ξεμπερδεύουμε άμεσα. Όμως, το κόστος και η χαμηλή ακρίβεια που προσφέρει, μας έσπρωξαν να φτιάξουμε τον δικό μας αισθητήρα!

Η λύση στο πρόβλημά μας είναι αρκετά απλή και το κυριότερο εξάρτημα που θα χρειαστούμε είναι ένα photocell, ή αλλιώς φωτοκύτταρο[3]. Πρόκειται για μία αντίσταση

1. Για την ιστορία, ο πιο προσιτός αισθητήρας για τη χρωματική ρύθμιση ενός monitor στοιχίζει γύρω στα 150 δολάρια, με αρκετά μοντέλα να ξεπερνούν το χιλιάρικο!
2. Μήπως έφτασε η ώρα να σχεδιάσουμε κι ένα Rubic's Cube solver bot; Χμ...
3. Τα ίδια εξαρτήματα ονομάζονται και LDR (Light-Dependent Resistors) ή Photoresistors. Το γιατί θα φανεί σύντομα ;)



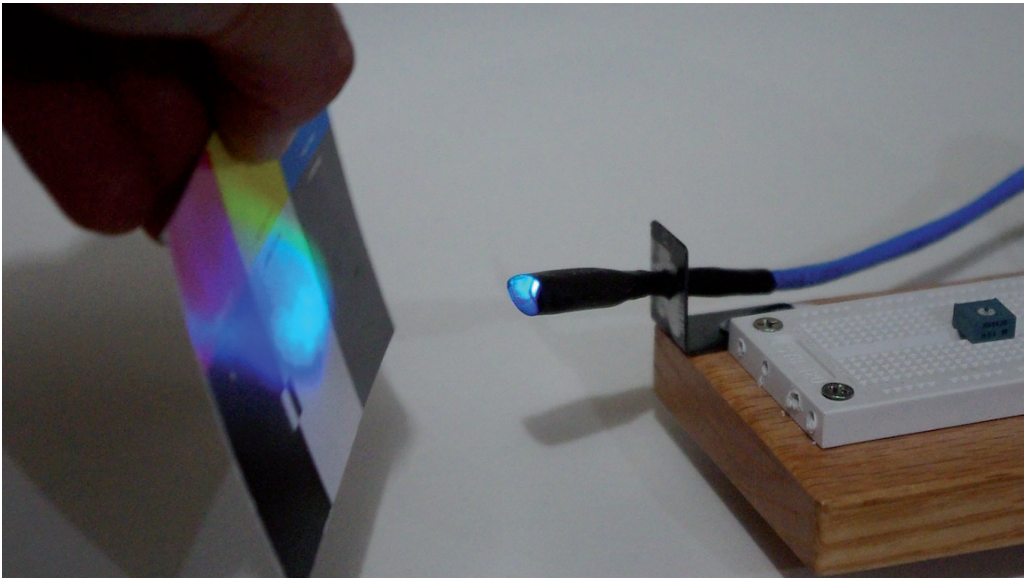
Τα φωτοκύτταρα (photocells) είναι μεταβλητές αντιστάσεις που αλλάζουν την τιμή τους, ανάλογα με την ένταση του φωτός. Ακριβώς αυτό το χαρακτηριστικό θα εκμεταλλευτούμε για να εντοπίζουμε τα χρώματα διαφόρων αντικειμένων.

που η τιμή της μεταβάλλεται ανάλογα με την ένταση του φωτός που πέφτει επάνω της. Έτσι, για να εκτιμήσουμε την ένταση του φωτός, αρκεί να μετρήσουμε την αντίσταση του photocell. Όπως καταλαβαίνετε, τα συγκεκριμένα εξαρτήματα δεν αποτελούν κάποιον αισθητήρα ακριβείας που έχει περάσει από δοκιμές ή/και κάποιου είδους "ρύθμιση" (calibration). Με άλλα λόγια, η *γενική* συμπεριφορά των photocells είναι πάντοτε ίδια, αλλά η απόκρισή τους (οι *συγκεκριμένες* μεταβολές που παρουσιάζουν στην αντίστασή τους) παρουσιάζει αποκλίσεις. Μεταβάλλοντας το φως που πέφτει πάνω σε δύο φωτοκύτταρα, θα παρατηρήσουμε παρόμοιες μεταβολές στις αντιστάσεις τους, αλλά οι τιμές τους θα είναι κατά πάσα πιθανότητα διαφορετικές!

Πριν προχωρήσουμε, επιτρέψτε μας να πούμε δυο λόγια για το πώς βλέπουμε εμείς[4], οι άνθρωποι: Τα μάτια μας περιλαμβάνουν χιλιάδες μικροσκοπικούς αισθητήρες, μερικοί από τους οποίους αντιλαμβάνονται την ένταση του φωτός (τα λεγόμενα ραβδία) και κάποιοι άλλοι τη συχνότητά του — ή πιο απλά το χρώμα (τα λεγόμενα κωνία). Αυτό που αντιλαμβανόμαστε σαν εικόνα, δηλαδή η πληροφορία σχετικά με την ένταση αλλά και το χρώμα του φωτός στις διάφορες περιοχές, προκύπτει από την αντανάκλαση του φωτός στα αντικείμενα. Κοιτώντας μια κόκκινη μπάλα, βλέπουμε τις ακτίνες φωτός που *δεν* απορροφήθηκαν από τη μπάλα και οι οποίες έχουν τη συχνότητα που αντιλαμβανόμαστε ως κόκκινο. Ας επιστρέψουμε τώρα στα φωτοκύτταρα. Τις περισσότερες φορές χρησιμοποιούνται σε απλές συσκευές, οι οποίες ενεργοποιούνται αυτόματα όταν μειώνεται ο φωτισμός. Όπως καταλαβαίνετε, λοιπόν, είναι κατασκευασμένα για να αντιδρούν στην ίδια περιοχή του φάσματος της ηλεκτρομαγνητικής ακτινοβολίας που αντιδρούν και τα μάτια μας: στο ορατό φως. Στο κάτω-κάτω, θα ήταν περίεργο αν το λαμπάκι νύχτας άναβε όταν σβήναμε το WiFi ή τον φούρνο μικροκυμάτων, έτσι δεν είναι; :D Μάλιστα, η συμπεριφορά των photocells εξαρτάται τόσο από την ένταση του φωτός, όσο και από τη συχνότητά του. Τελικά, η συμπεριφορά τους είναι παρόμοια με εκείνη των ματιών μας.

Όπως υποψιάζεστε, μπορούμε να χρησιμοποιήσουμε ένα photocell όχι μόνο για την ανίχνευση φωτός (γενικά), αλλά και για την ανίχνευση χρωμάτων. Το μόνο που έχουμε να κάνουμε είναι να φωτίσουμε ένα αντικείμενο με φως συγκεκριμένου χρώματος και να μετρήσουμε την ένταση της αντανάκλασής του. Αν το φωτοκύτταρο "δηλώσει" ότι η αντανάκλαση είναι ισχυρή, σημαίνει ότι το φως δεν απορροφήθηκε κι άρα το αντικείμενο έχει το χρώμα του φωτός που εκπέμψαμε! Αν πραγματοποιήσουμε την ίδια διαδικασία τρεις φορές, με κόκκινο, πράσινο και μπλε φως, τότε με τη βοήθεια του photocell μπορούμε να εκτιμήσουμε το ποσοστό του κόκκινου, του πράσινου και του μπλε χρώματος, που έχει το αντικείμενο. Το πιάσατε; Πραγματοποιώντας μία μέτρηση με καθένα από αυτά τα χρώματα, μπορούμε να εκτιμήσουμε το χρώμα ενός αντικειμένου λαμβάνοντας μια περιγραφή τύπου RGB.

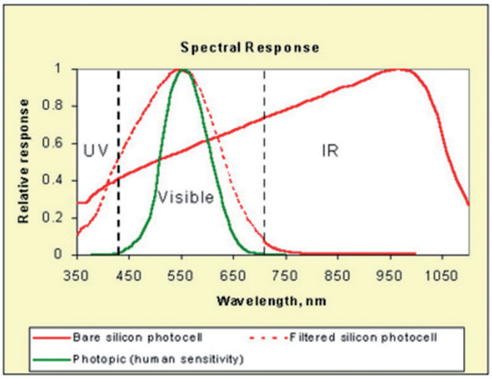
4. Αν είσαστε cyborg, συνεχίστε με την επόμενη παράγραφο!



Μια απλή κατασκευή που εντοπίζει τα χρώματα των αντικειμένων που βρίσκονται μπροστά της. Κάτι τέτοιο θα κατασκευάσουμε κι εμείς...

Συγκέντρωση υλικών

Προφανώς το βασικότερο εξάρτημα είναι το φωτοκύτταρο, το οποίο μπορούμε είτε να αγοράσουμε είτε να αφαιρέσουμε από κάποια συσκευή (ένα φωτάκι νύχτας δωματίου, ένα φωτάκι νύχτας κήπου κ.ά). Φυσικά, η καλύτερη λύση θα ήταν να το αγοράσουμε, καθώς ο πωλητής ενδέχεται να παρέχει κι ένα datasheet με τα χαρακτηριστικά και, κυρίως, με την απόκριση του συγκεκριμένου photocell σε κάθε χρώμα. Ωστόσο, επειδή δεν απαιτούμε μεγάλη ακρίβεια, ένα οποιοδήποτε photocell θα είναι αρκετό για την κατασκευή μας. Πριν πιάσουμε το κατσαβίδι κι αρχίσουμε το ψάξιμο σε παλιές συσκευές, να σημειώσουμε ότι τα 20 photocells στοιχίζουν μόλις 0,94€ στο eBay. Το δικό μας θύμα ήταν ένα φωτάκι κήπου — και δηλώνουμε ένοχοι για ό,τι του συνέβη :D



Τα φωτοκύτταρα αντιδρούν στην ίδια περιοχή του ηλεκτρομαγνητικού φάσματος, όπου αντιδρά και το ανθρώπινο μάτι.

Για τον φωτισμό του εκάστοτε αντικειμένου (πώς αλλιώς θα μετρήσουμε την αντανάκλαση;) χρειαζόμαστε τρία LEDs: ένα κόκκινο, ένα πράσινο κι ένα μπλε. Η απλούστερη λύση, όμως, είναι να χρησιμοποιήσουμε ένα RGB LED. Τέτοια LEDs έχουμε χρησιμοποιήσει και σε παλιότερες κατασκευές, οπότε δεν θα χρειαστεί να καταφύγουμε σε κάποια αγορά. Για να προστατέψουμε το LED θα χρειαστούμε και τρεις αντιστάσεις των 470Ω, αν και θα μπορούσαμε να χρησιμοποιήσουμε οποιαδήποτε άλλη, παραπλήσια τιμή. Εμείς επιλέξαμε τα 470Ω, επειδή διαθέταμε ήδη αυτές τις αντιστάσεις. Τέ-



Μπορούμε να "δανειστούμε" ένα photocell από ένα λαμπάκι νύχτας ή καλύτερα από ένα φως κήπου. Επιλέξαμε να πειράξουμε ένα φως κήπου, γιατί μαζί με τον αισθητήρα πήραμε και άλλα εξαρτήματα, όπως επαναφορτιζόμενες μπαταρίες και ηλιακά πάνελ :D

λος, θα χρειαστούμε μια αντίσταση των 10ΚΩ. Αναρωτιέστε τι θα την κάνουμε; Για την μέτρηση της αντίστασης του photocell θα φτιάξουμε έναν μικρό διαιρέτη τάσης: Θα συνδέσουμε το photocell σε σειρά με την αντίσταση, κι αυτό το απλό "δικτύωμα" θα συνδεθεί μεταξύ 5V και GND. Με τη βοήθεια του Arduino θα μετράμε την τάση που δίνει ο διαιρέτης κι από αυτήν θα υπολογίζουμε την αντίσταση του photocell.

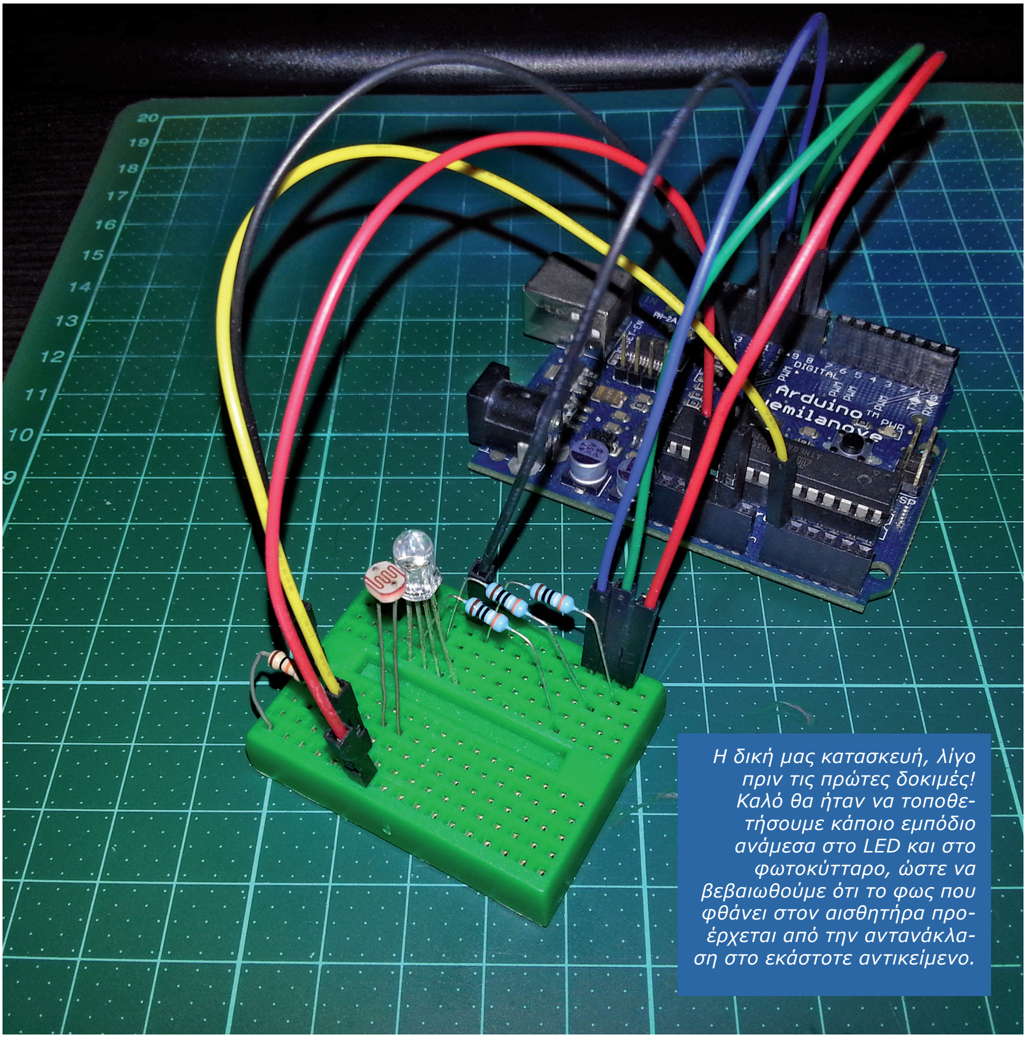
Το μαγείρεμα

Ξεκινάμε με το κύκλωμα. Οι τρεις ακροδέκτες του RGB LED που ελέγχουν τα επιμέρους χρώματα, συνδέονται μέσω μιας αντίστασης των 470Ω με τα pin 9, 10 και 11 του Arduino. Ο τέταρτος ακροδέκτης του RGB LED (ο λεγόμενος "κοινός") συνδέεται στο GND[5]. Ο ένας ακροδέκτης του φωτοκύτταρου συνδέεται στα +5V, ενώ ο άλλος ακροδέκτης συνδέεται με το Analog pin 0 του Arduino και με την αντίσταση των 10ΚΩ. Τέλος, το άλλο άκρο της αντίστασης συνδέεται στο GND, ώστε να σχηματιστεί ο διαιρέτης τάσης που περιγράψαμε νωρίτερα.

Το πρόγραμμα στο Arduino (`http://bit.ly/dh027colorpicker`) ξεκινάει ρυθμίζοντας τα αντίστοιχα pins σαν εξόδους, ενώ ενεργοποιεί και τη σειραϊκή σύνδεση, μέσω της οποίας θα στέλνει το χρώμα που εντοπίζει. Ο κυρίως κώδικας πραγματοποιεί αρχικά μια απλοϊκή αλλά χρήσιμη ρύθμιση (calibration): Εξετάζει τη συμπεριφορά του photocell που διαθέτουμε, μετρώντας τα άκρα των τιμών που παρέχει. Αυτό επιτυγχάνεται με μία μέτρηση για το μαύρο και με μία για πλήρη ένταση του φωτός. Μάλιστα, αυτό το σετάκι μετρήσεων επαναλαμβάνεται μία φορά για το κόκκινο, μία για το πράσινο και μία για το μπλε χρώμα. Οι τιμές που προκύπτουν αποθηκεύονται σε πίνακες και χρησιμοποιούνται αργότερα για την κανονικοποίηση (normalization) των υπόλοιπων μετρήσεων.

Όλη η μαγεία εντοπίζεται στη συνάρτηση checkColor. Εκεί, για κάθε χρώμα του RGB LED παίρνονται 5 διαφορετικές μετρήσεις, από τις οποίες κρατάμε τον μέσο όρο. Εδώ χρειάζεται λίγη προσοχή: Αναρωτιέστε γιατί πραγματοποιούμε μια παύση του μισού δευτερολέπτου, αφότου ανάψουμε το LED και μέχρι να πάρουμε τη μέτρηση; Τα φωτοκύτταρα παρουσιάζουν μεγάλο χρόνο απόκρισης και, με απλά λόγια, η τιμή της αντίστασής τους μεταβάλλεται με αργό ρυθμό όταν αλλάζει η ένταση του φωτός. Για να ολοκληρωθεί ο υπολογισμός κάθε επιμέρους χρώματος (κόκκινο, πράσινο, μπλε),

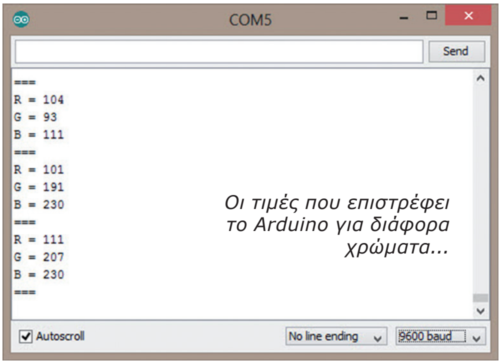
5. Τα περισσότερα RGB LEDs είναι τύπου CC ή Common Cathode. Έτσι, ο τέταρτος ακροδέκτης τους (ο κοινός) συνδέεται εσωτερικά στις καθόδους των τριών επιμέρους LED. Αν το δικό σας RGB LED είναι τύπου CA (Common Anode) θα πρέπει να συνδέσετε τον κοινό ακροδέκτη στα 5V. Βέβαια, αυτή η αλλαγή προϋποθέτει αλλαγές και στον κώδικα που θα δούμε αργότερα. Για λόγους απλότητας, θα σας προτείναμε να πάρετε ένα RGB LED τύπου CC.



Η δική μας κατασκευή, λίγο πριν τις πρώτες δοκιμές! Καλό θα ήταν να τοποθετήσουμε κάποιο εμπόδιο ανάμεσα στο LED και στο φωτοκύτταρο, ώστε να βεβαιωθούμε ότι το φως που φθάνει στον αισθητήρα προέρχεται από την αντανάκλαση στο εκάστοτε αντικείμενο.

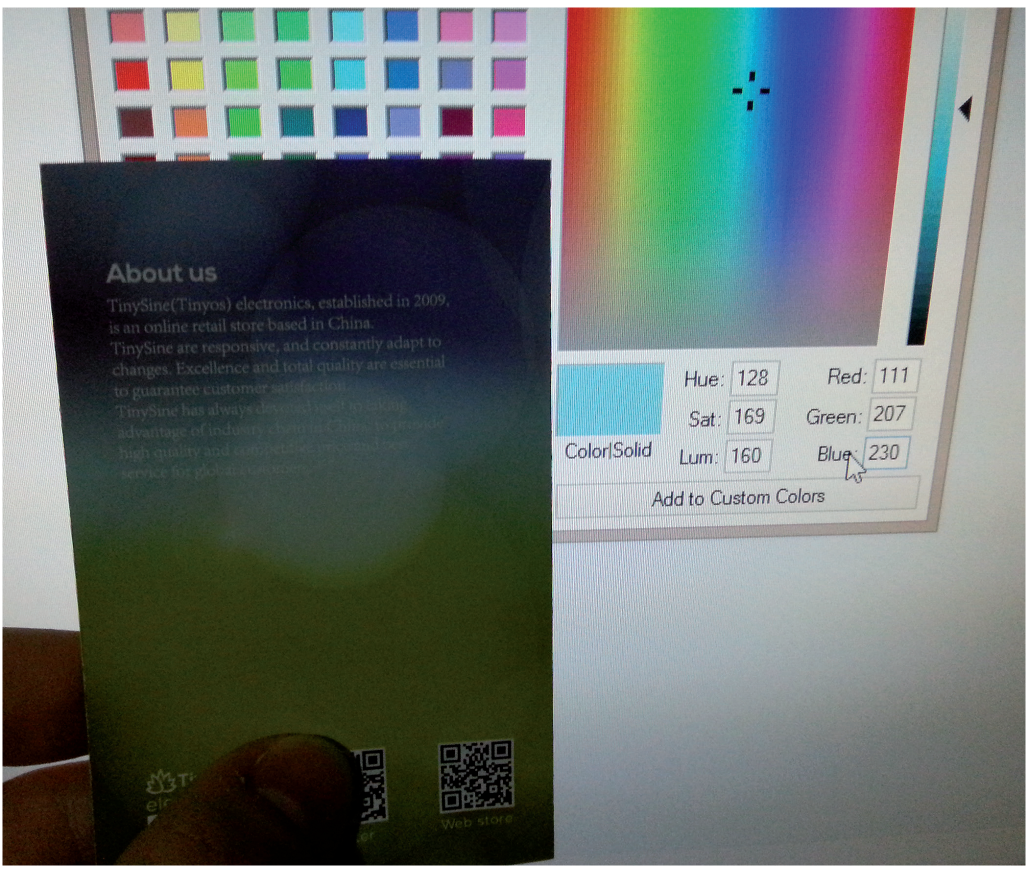
αφαιρούμε από το μέσο όρο των πέντε μετρήσεων την τιμή που μετρήσαμε κατά το calibration, στο πλήρες σκοτάδι. Στη συνέχεια, διαιρούμε το αποτέλεσμα αυτής της πράξης με τη διαφορά των δύο "ακραίων" τιμών για το αντίστοιχο χρώμα (τιμή σε πλήρη φωτισμό – τιμή στο σκοτάδι). Με αυτόν τον τρόπο επιτυγχάνεται η κανονικοποίηση που αναφέραμε προηγουμένως. Έτσι, για κάθε χρώμα, λαμβάνουμε μια τιμή που βρίσκεται στο διάστημα από 0 έως 1. Καθεμία από αυτές τις τιμές πολλαπλασιάζεται με το 255 και τελικά, για κάθε χρώμα παίρνουμε μια τιμή στο διάστημα 0 έως 255! Με άλλα λόγια παίρνουμε την περιγραφή του χρώματος σε RGB.

Το μόνο που μένει είναι να τυπώσουμε τα αποτελέσματα και να προχωρήσουμε στην επόμενη ανίχνευση! Νομίζουμε ότι η λογική του προγράμματός μας είναι αρκετά απλή. Ωστόσο, θα πρέπει να επιστήσουμε την προσοχή σας σε δύο μεγέθη που ενδεχομένως θα χρειαστεί να τροποποιήσετε, για να πάρετε τα καλύτερα δυνατά αποτελέσματα. Πρώτον, πόσες μετρήσεις θα πρέπει να παίρνουμε για καθένα από



Οι τιμές που επιστρέφει το Arduino για διάφορα χρώματα...

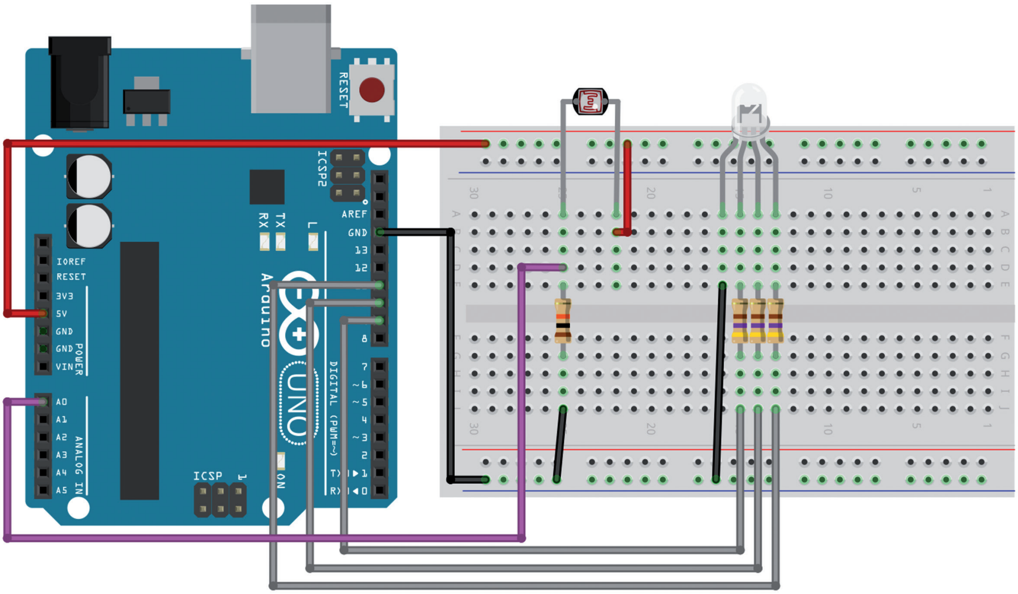
τα τρία χρώματα; Η παράμετρος αυτή πιθανώς να μην παίζει μεγάλο ρόλο, αν το φωτοκύτταρο ανταποκρίνεται άμεσα στις αλλαγές. Στις δικές μας δοκιμές, πάντως, διαπιστώσαμε ότι οι πέντε μετρήσεις είναι αρκετές, ενώ με λιγότερες τα αποτελέσματα των πειραμάτων δεν ήταν σταθερά (δεν παρουσίαζαν επαναληψιμότητα). Δεύτερον, πρέπει να ορίσουμε το σωστό χρόνο αναμονής, από τη στιγμή που ανάβουμε το LED μέχρι να πάρουμε τη μέτρηση. Αν έχουμε στα χέρια μας κάποιο datasheet θα πρέπει να το συμβουλευτούμε, μιας και σίγουρα θα αναφέρει πόσο χρόνο απαιτεί το photocell για να σταθεροποιηθεί στις νέες συνθήκες. Αν όχι, ο μόνος τρόπος είναι οι δοκιμές!



Θα μπορούσαμε να βρούμε μια μονόχρωμη κάρτα αλλά μας αρέσουν τα δύσκολα — χωρίς να μας απασχολεί το ενδεχόμενο της αποτυχίας! Αυτή τη φορά πάντως δεν αποτύχαμε :) Ο αισθητήρας εντόπισε το χρώμα επιτυχώς. Για να δούμε το χρώμα στον υπολογιστή, χρησιμοποιήσαμε το εργαλείο χρωμάτων του παλιού, καλού Paint των Windows.

Αποτίμηση

Τα αποτελέσματα είναι αρκετά εντυπωσιακά, ειδικά αν σκεφτούμε το συνολικό κόστος της κατασκευής. Ο αισθητήρας μας κατάφερε να εντοπίσει με μεγάλη ακρίβεια πολλά και διαφορετικά χρώματα αντικειμένων. Προφανώς, με έναν εξειδικευμένο αισθητήρα, προ-ρυθμισμένο απ' την κατασκευάστρια εταιρεία, θα παίρναμε πιο ακριβείς μετρήσεις. Ωστόσο, για τις περισσότερες δουλειές η κατασκευή μας προσφέρει ικανοποιητικές τιμές. Ένα προβληματάκι που παρατηρήσαμε ήταν ότι ο αισθητήρας επηρεαζόταν αρκετά από το είδος των επιφανειών (από το αν ήταν γυαλιστερές ή ματ). Όταν μετρούσαμε χρώματα πάνω σε χαρτί τα αποτελέσματα ήταν πολύ καλά, ενώ όταν δοκιμάζαμε σε γυαλιστερά πλαστικά, γυαλιά ή μέταλλα, οι μετρήσεις ήταν αλλοπρόσαλλες. Για την ώρα, πάντως, είμαστε αρκετά ευχαριστημένοι. Ελπίζουμε να ικανοποιηθείτε κι εσείς από τα αποτελέσματα και περιμένουμε να ακούσουμε τα σχόλιά σας. Φυσικά, θα χαρούμε να μάθουμε και τις εφαρμογές που σκεφτήκατε για τον αισθητήρα χρωμάτων.



Το κύκλωμα στο Fritzing! Αρκετά απλό στη λογική του και με ελάχιστα εξαρτήματα, που κατά πάσα πιθανότητα διαθέτουν ήδη όσοι ασχολούνται με τα ηλεκτρονικά.

Skill: Advanced
Tags: pfSense, OpenBSD, bridging



Bridging στο BSD για routers κατηγορίας kick-ass!

Μετά το εισαγωγικό άρθρο στο bridging, που αρχίζει από τη σελίδα 40, είναι πλέον ώρα να δούμε πώς γεφυρώνουμε network interfaces προκειμένου να φτιάχνουμε routers που καλύπτουν όσο το δυνατόν καλύτερα τις ανάγκες μας. Αν δεν έχετε το hardware για να παίξετε, δεν υπάρχει κανένα πρόβλημα: Μπορείτε να ξεκινήσετε τους πειραματισμούς σας σε εικονικό περιβάλλον — και σας δείχνουμε πώς.

by sub2raw

Σκοπεύετε λοιπόν να φτιάξετε τον δικό σας router για το σπίτι, για το γραφείο, για κάποιο εργαστήριο στο σχολείο ή στο πενεπιστήμιο κ.ο.κ. Θέλουμε κατ' αρχάς να σας πούμε ότι ζηλεύουμε λίγο, αφού ακόμη κι αν γνωρίζετε ακριβώς πώς θα κινηθείτε και τι θα κάνετε, θα το διασκεδάσετε ούτως ή άλλως. Θα θέλαμε εξάλλου να υπογραμμίσουμε ότι το bridging είναι πολύ πιθανό να σας γλυτώσει από περιττά έξοδα ή/και να σας βοηθήσει σε θέματα ασφάλειας κι επιδόσεων. Αν τώρα μας ρωτήσετε, για λειτουργικό σύστημα του router προτείνουμε να επιλέξετε μεταξύ των pfSense κι OpenBSD. Στο παρόν άρθρο, μάλιστα, δείχνουμε πώς φτιάχνουμε bridges σε καθένα εξ αυτών των συστημάτων.

Τώρα, από τη στιγμή που συζητάμε για τον δικό μας, DIY router, να πούμε δυο λόγια και για το hardware. Μεταξύ των επιλογών που έχουμε υπάρχουν και λύσεις σαν αυτές που προσφέρει η Soekris. Δείτε για παράδειγμα τη σειρά net6501, στο http://soekris.eu/shop/net6501_en. Το καλό με το Soekris hardware είναι ότι συνεργάζεται άψογα με το *BSD, φυσικά και με το Linux. Είναι επίσης χαμηλής ενεργειακής κατανάλωσης, εντελώς αθόρυβο αλλά και "μαζεμένο", υπό την έννοια ότι καταλαμβάνει λίγο χώρο. Ένα "μείον" του Soekris hardware είναι το κόστος του, το οποίο πιθανώς θα βρείτε τσιμπημένο (εμείς τουλάχιστον αυτό σκεφτήκαμε). Μιλάμε όμως για εξειδικευμένες λύσεις και συγκριτικά ισχυρό hardware, επομένως εννοείται ότι οι τιμές θα 'ναι πάνω από εκείνες των τυπικών, consumer grade συσκευών.

Εναλλακτικά, ίσως αποφασίσετε να μετατρέψετε σε router έναν σχετικά παλιό υπολογιστή, ο οποίος φαίνεται παροπλισμένος αλλά στην πραγματικότητα έχει πολλά να προσφέρει ακόμα. Αν βγείτε στην αγορά για μία ή περισσότερες κάρτες Ethernet —το box χρειάζεται τουλάχιστον δύο— προτιμήστε εκείνες της Intel, αφού υποστηρίζονται άριστα από το *BSD (και ναι, κι από το Linux). Καλού κακού, πάντως, πριν την αγορά ρίξτε μια καλή ματιά στη σχετική λίστα με το υποστηριζόμενο hardware — εννοείται από πλευράς του OS που σκοπεύετε να χρησιμοποιήσετε. Τέλος, μια καλή ιδέα είναι ν' απαλλάξετε το box από τον κλασικό, μαγνητικό του δίσκο και να του χαρίσετε έναν μικρό SSD. Αυτομάτως το box θα γίνει πιο αθόρυβο, θα καταναλώνει λιγότερη ενέργεια, ενώ θα έχετε και χώρο για να στήσετε έξτρα εφαρμογές, όπως, π.χ., web proxy cache ή/και κάποιο intrusion detection system.

Ας περάσουμε όμως στο θέμα μας, που είναι το bridging.

Bridging με το pfSense

Το pfSense είναι το αγαπημένο λειτουργικό σύστημα όλων εκείνων που θέλουν να δημιουργήσουν το δικό τους, πανίσχυρο router/firewall, στο οποίο μάλιστα θα μπορούν να προσθέσουν κι ένα σωρό δυνατότητες κι εφαρμογές όπως, π.χ., transparent proxy, IDS κ.ο.κ. Για μια αναλυτική παρουσίαση της εγκατάστασης, της ρύθμισης αλλά και της χρήσης του pfSense, προτείνουμε να παρακολουθήσετε το 7ο επεισόδιο της 1ης σεζόν του deltaCast:

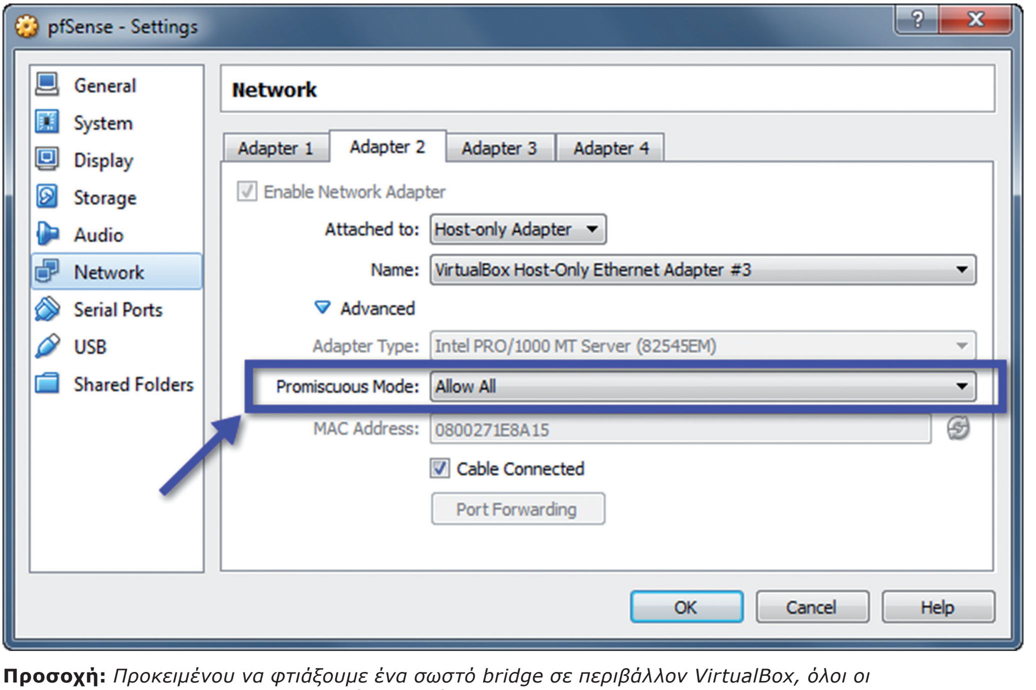
`http://deltahacker.gr/deltacast-s01e07`

Δουλεύοντας τώρα σε ένα Virtualbox VM, το οποίο διαθέτει τέσσερις εικονικούς network adapters, θα δούμε πώς φτιάχνουμε ένα bridge interface που γεφυρώνει τους τρεις από τους τέσσερις, δηλαδή αυτούς που βλέπουν οι πελάτες του pfSense. Ο τέταρτος adapter είναι για τη σύνδεση WAN και θα μείνει εκτός bridge. Πριν προχωρήσουμε, οφείλουμε να δούμε καλύτερα τους adapters του VM μας.

Adapter 1. Είναι γεφυρωμένος από το Virtualbox με τον αληθινό Ethernet adapter του host computer, ο οποίος είναι συνδεδεμένος ενσύρματα με τον αληθινό (VDSL) router του τοπικού δικτύου. Πρόκειται για το WAN interface του router και για το λειτουργικό του σύστημα είναι το em0 interface.

Adapter 2. Είναι τύπου host-only, έχει τον ρόλο του LAN interface για τον router, ενώ για το λειτουργικό του σύστημα είναι το em1 interface. Μέσω αυτού του adapter συνδεόμαστε στο web panel του pfSense χρησιμοποιώντας κάποιον web browser, ο οποίος τρέχει είτε στο host computer είτε σε ένα από τα VMs *πίσω* από το pfSense — αρκεί βεβαίως να 'ναι συνδεδεμένα στο LAN interface του προαναφερθέντος. Πρακτικά, για το εικονικό περιβάλλον στο οποίο εργαζόμαστε, αυτό σημαίνει ότι τα VM-πελάτες βγαίνουν στον έξω κόσμο μέσω network adapter που είναι κι αυτός host-only — και μάλιστα ανήκει στο ίδιο host-only network όπου ανήκει κι ο Adapter 2. Διαφορετικά: Αν το όνομα του host-only adapter που αντιστοιχεί στο LAN interface είναι το "VirtualBox Host-Only Ethernet Adapter #3" (βλ. στην κατηγορία Network, στις ρυθμίσεις του pfSense VM), τότε για να πούμε ότι ένα οποιοδήποτε VM είναι πελάτης του pfSense, αυτό πρέπει να βγαίνει στον έξω κόσμο μέσω host-only adapter με όνομα επίσης το "VirtualBox Host-Only Ethernet Adapter #3". Παρατηρήστε και τις ρυθμίσεις δικτύωσης της εκάστοτε εικονικής μηχανής, και η περιγραφή που μόλις επιχειρήσαμε ξαφνικά θ' αποκτήσει νόημα :)

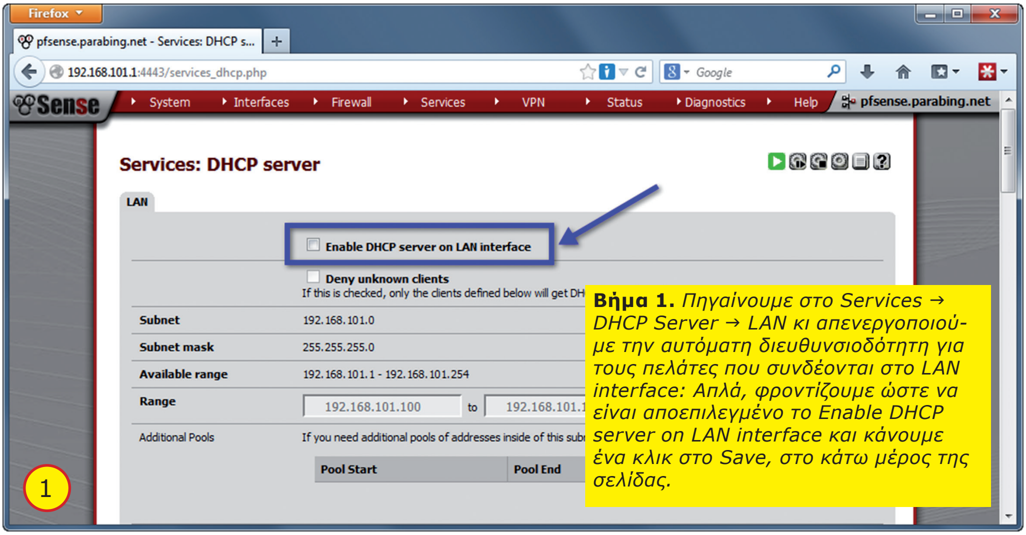
Adapter 3 & **Adapter 4.** Είναι κι αυτοί τύπου host-only, ανήκουν σε διαφορετικά host-only networks απ' αυτό που ανήκει ο Adapter 2, εξομοιώνουν άλλα δύο Ethernet ports του router, σε λίγο το pfSense θα τα ονομάζει OPT1 και OPT2, ενώ για το λειτουργικό σύστημα είναι τα em2 και em3 interfaces. Μετά την εγκατάσταση του pfSense δεν έχουν IP — και μάλιστα δεν είναι καν ενεργά.



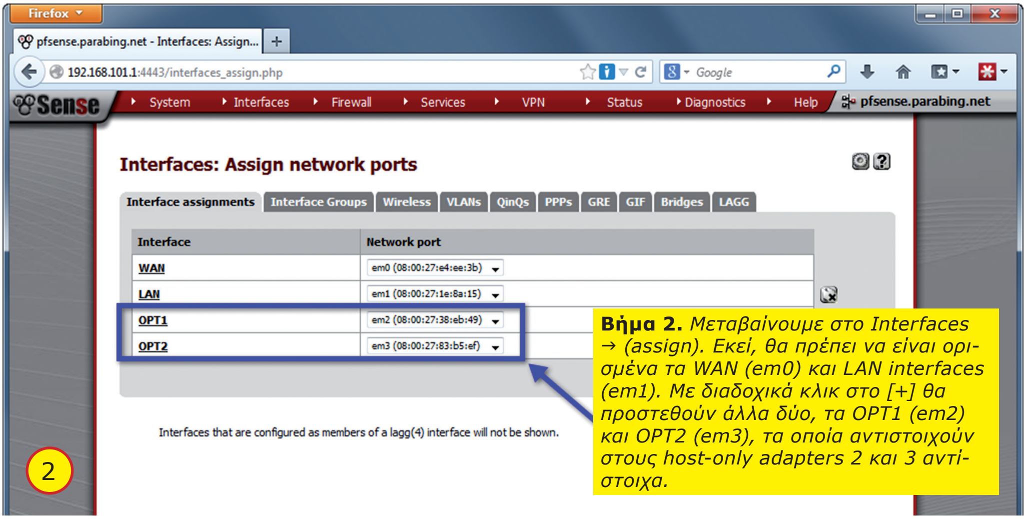
Προσοχή: *Προκειμένου να φτιάξουμε ένα σωστό bridge σε περιβάλλον VirtualBox, όλοι οι network adapters που θα γεφυρωθούν θα πρέπει να 'ναι σε Promiscuous Mode.*

Δουλειά μας για τη συνέχεια είναι να ενεργοποιήσουμε τα OPT1, OPT2 και να τα γεφυρώσουμε με το LAN (η σειρά δεν έχει σημασία). Διαφορετικά: Αυτό που θα κάνουμε είναι να φτιάξουμε ένα bridge group με μέλη τα OPT1, OPT2 και LAN. Βεβαίως, θα χρειαστεί να ρυθμίσουμε κατάλληλα τον DHCP server και τον DNS forwarder, καθώς και να προσθέσουμε μερικούς απαραίτητους κανόνες στο pf firewall, ώστε να 'ναι δυνατή η αυτόματη διευθυνσιοδότηση αλλά και η επικοινωνία των clients μεταξύ τους, καθώς και με τον έξω κόσμο. Πρακτικά, αν είχαμε αληθινό hardware με τέσσερα Ethernet ports, εξαιρώντας εκείνο που θα συνδεόταν στο modem, σε οποιοδήποτε από τα άλλα συνδέαμε ένα μηχάνημα τότε αυτό θα έπαιρνε IP από το ίδιο subnet, θα έβλεπε άλλα μηχανήματα-πελάτες και θα έβγαινε στο Internet.

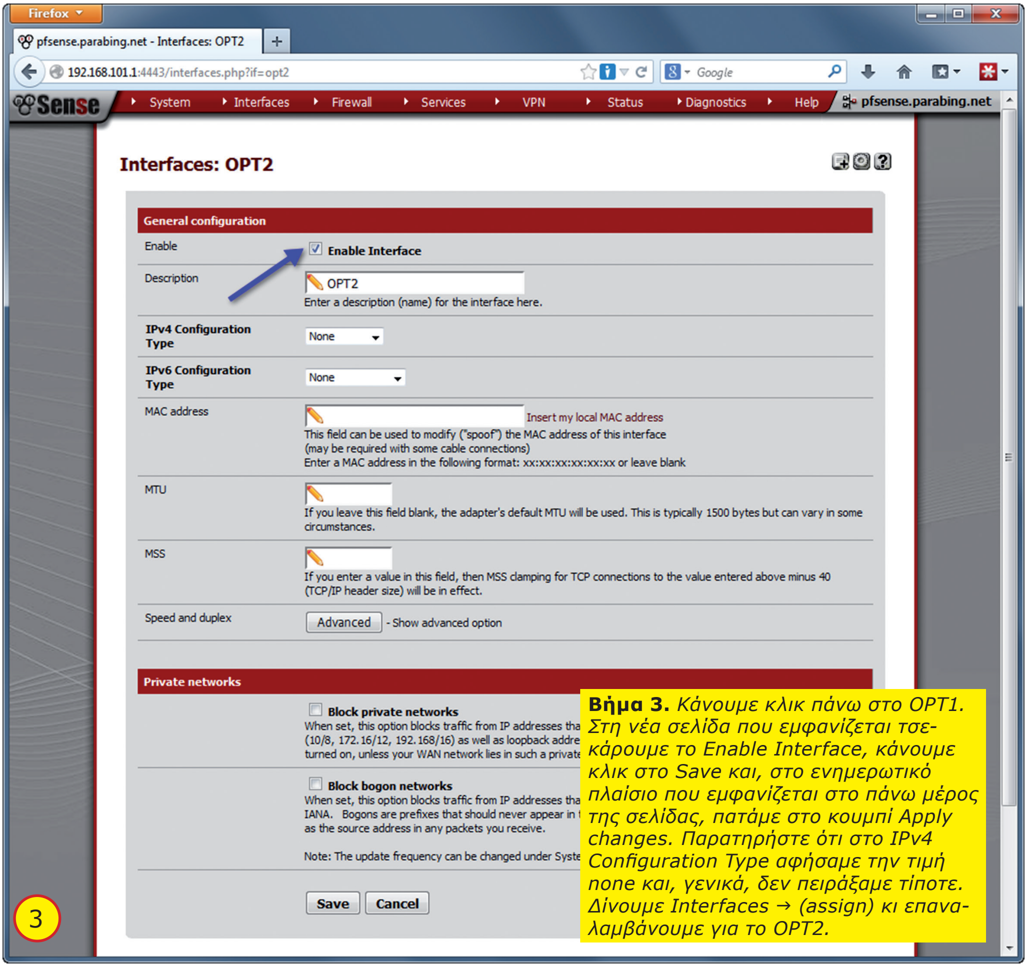
Δείχνουμε ακολούθως πώς εργαζόμαστε, αναλυτικά και βήμα προς βήμα, μέσα από το web panel του pfSense.



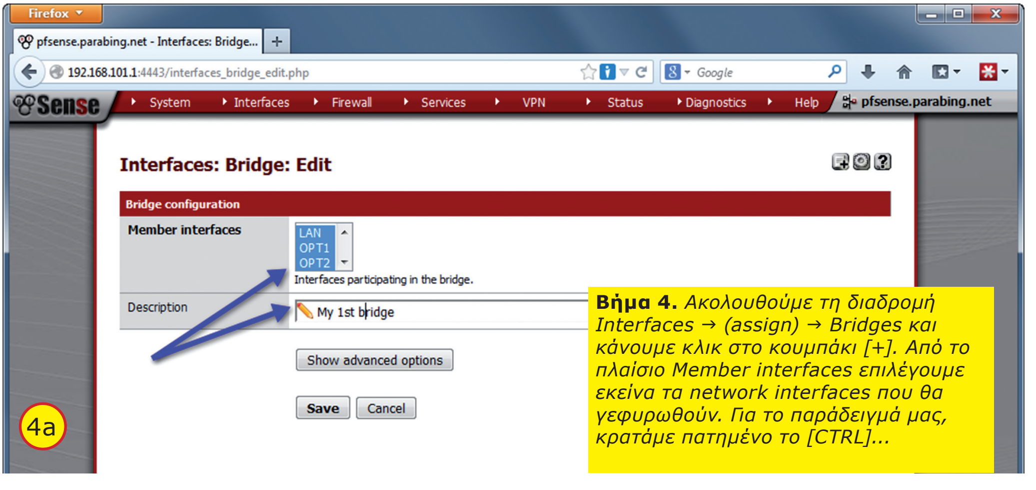
Βήμα 1. *Πηγαίνουμε στο Services → DHCP Server → LAN κι απενεργοποιούμε την αυτόματη διευθυνσιοδότητη για τους πελάτες που συνδέονται στο LAN interface: Απλά, φροντίζουμε ώστε να είναι αποεπιλεγμένο το Enable DHCP server on LAN interface και κάνουμε ένα κλικ στο Save, στο κάτω μέρος της σελίδας.*



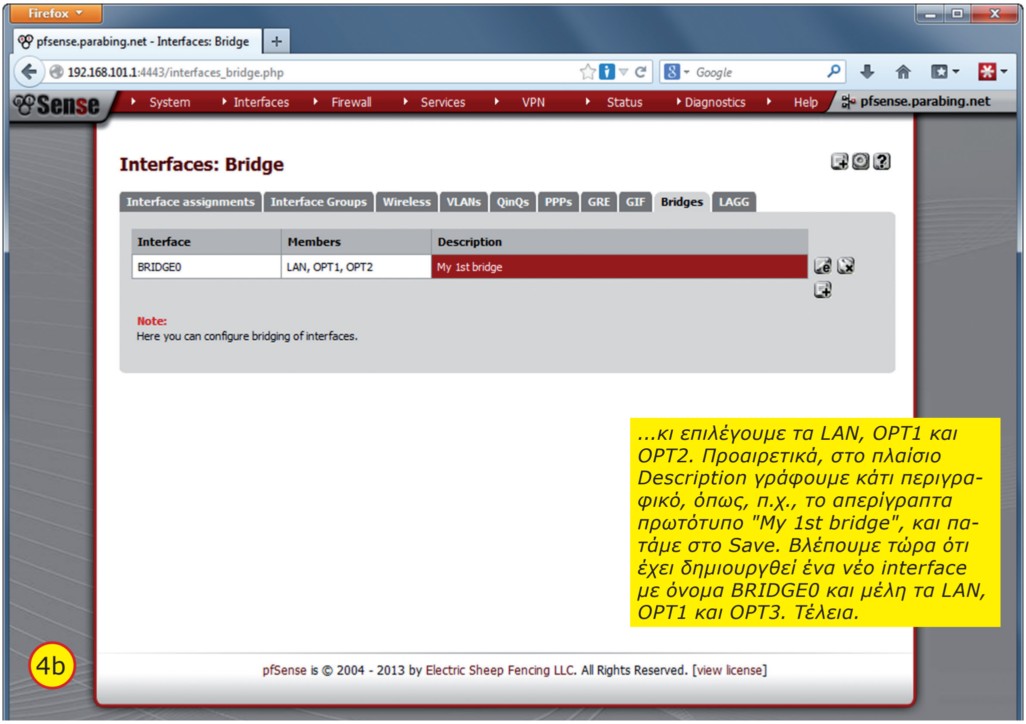
Βήμα 2. *Μεταβαίνουμε στο Interfaces → (assign). Εκεί, θα πρέπει να είναι ορισμένα τα WAN (em0) και LAN interfaces (em1). Με διαδοχικά κλικ στο [+] θα προστεθούν άλλα δύο, τα OPT1 (em2) και OPT2 (em3), τα οποία αντιστοιχούν στους host-only adapters 2 και 3 αντίστοιχα.*



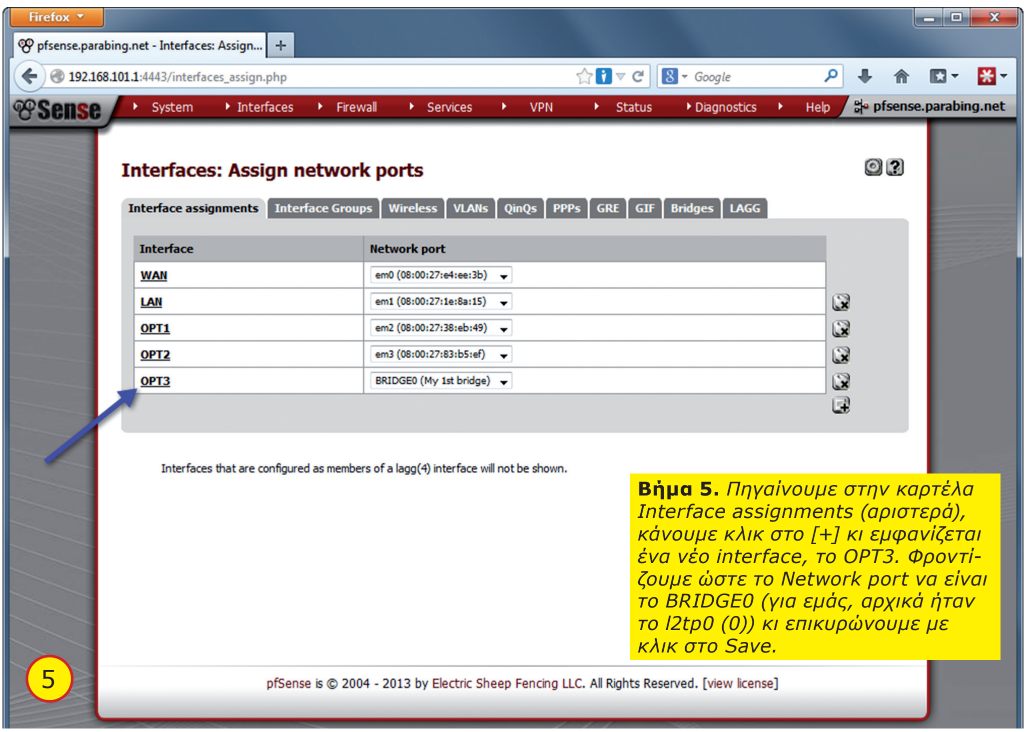
Βήμα 3. *Κάνουμε κλικ πάνω στο OPT1. Στη νέα σελίδα που εμφανίζεται τσεκάρουμε το Enable Interface, κάνουμε κλικ στο Save και, στο ενημερωτικό πλαίσιο που εμφανίζεται στο πάνω μέρος της σελίδας, πατάμε στο κουμπί Apply changes. Παρατηρήστε ότι στο IPv4 Configuration Type αφήσαμε την τιμή none και, γενικά, δεν πειράξαμε τίποτε. Δίνουμε Interfaces → (assign) κι επαναλαμβάνουμε για το OPT2.*



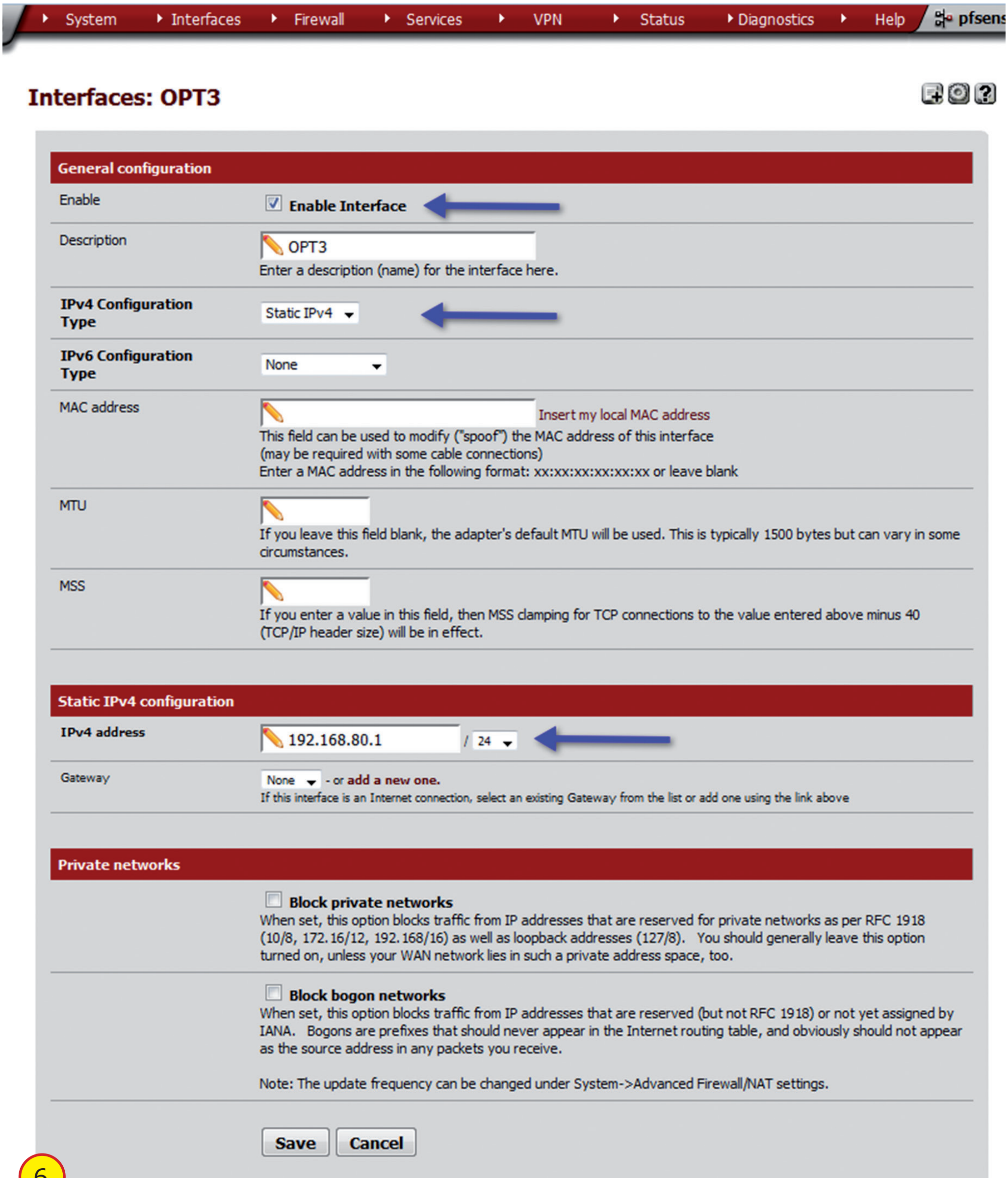
Βήμα 4. *Ακολουθούμε τη διαδρομή Interfaces → (assign) → Bridges και κάνουμε κλικ στο κουμπάκι [+]. Από το πλαίσιο Member interfaces επιλέγουμε εκείνα τα network interfaces που θα γεφυρωθούν. Για το παράδειγμά μας, κρατάμε πατημένο το [CTRL]...*



...κι επιλέγουμε τα LAN, OPT1 και OPT2. Προαιρετικά, στο πλαίσιο Description γράφουμε κάτι περιγραφικό, όπως, π.χ., το απερίγραπτα πρωτότυπο "My 1st bridge", και πατάμε στο Save. Βλέπουμε τώρα ότι έχει δημιουργηθεί ένα νέο interface με όνομα BRIDGE0 και μέλη τα LAN, OPT1 και OPT3. Τέλεια.

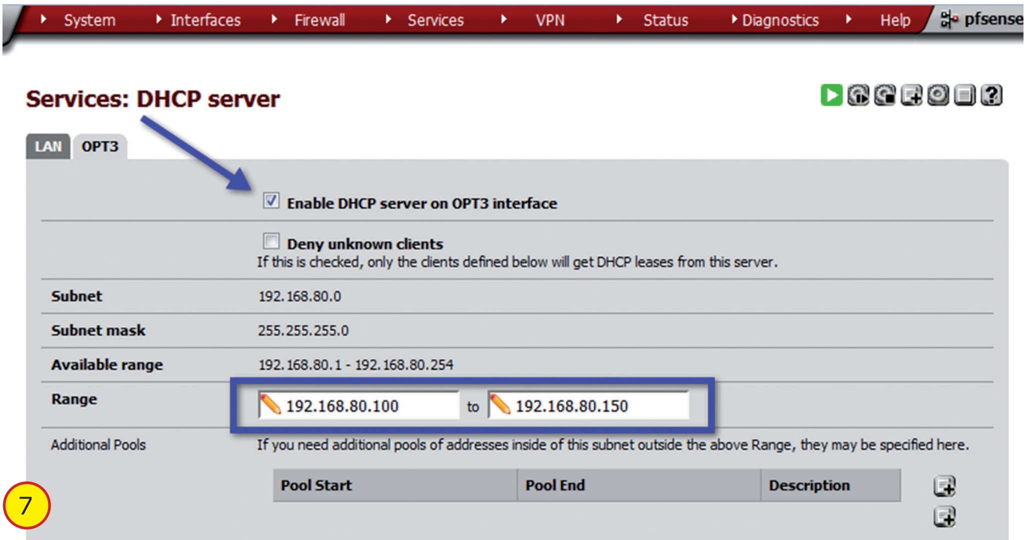


Βήμα 5. *Πηγαίνουμε στην καρτέλα Interface assignments (αριστερά), κάνουμε κλικ στο [+] κι εμφανίζεται ένα νέο interface, το OPT3. Φροντίζουμε ώστε το Network port να είναι το BRIDGE0 (για εμάς, αρχικά ήταν το l2tp0 (0)) κι επικυρώνουμε με κλικ στο Save.*

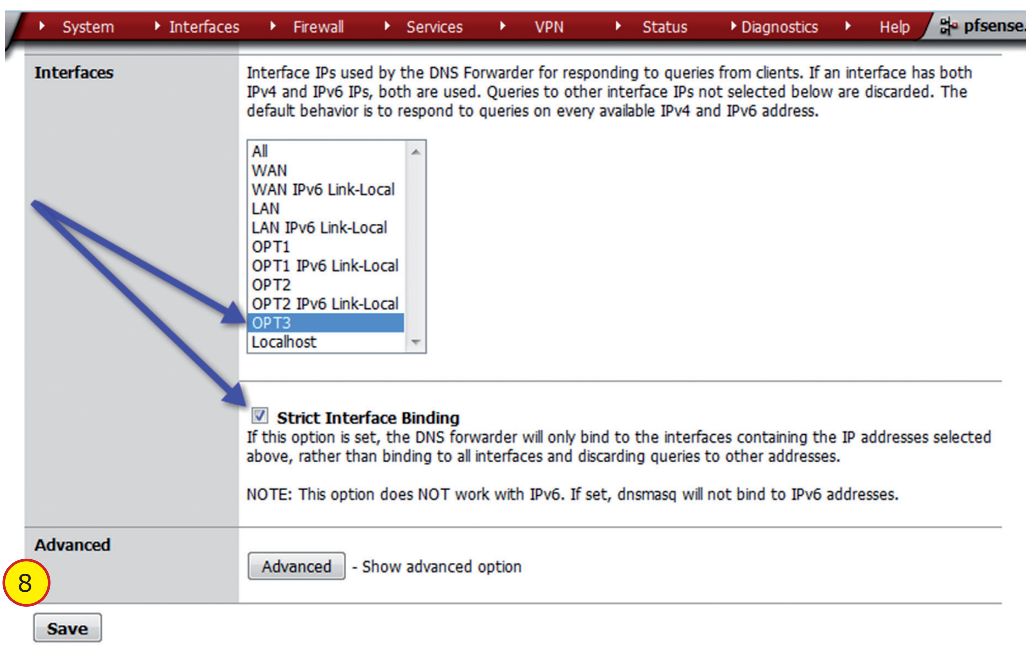


6

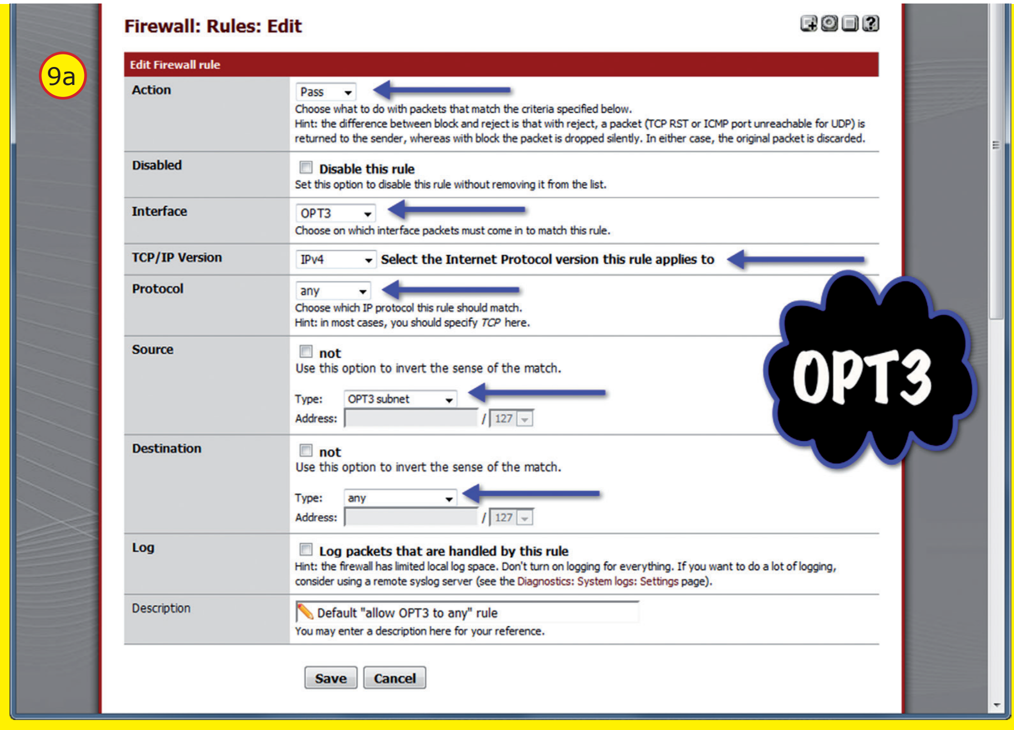
Βήμα 6. *Με κλικ πάνω στο OPT3, εμφανίζεται η σελίδα ρύθμισής του. Φυσικά, εκεί τσεκάρουμε το Enable Interface και στο IPv4 Configuration Type θέτουμε Static IPv4. Λίγο πιο κάτω στη σελίδα, στο πλαίσιο IPv4 address, πληκτρολογούμε μια non-routable διεύθυνση IP από ένα κατάλληλο για το setup μας subnet. Εμείς, π.χ., δώσαμε τη διεύθυνση 192.168.80.1, μιας και κανένα από τα host-only networks δεν ήταν το 192.168.80.0/24. Με την ευκαιρία, να σημειώσουμε ότι στα δεξιά της διεύθυνσης που θα δώσετε, δίπλα στο "/", φροντίστε να υπάρχει η τιμή 24 — εκτός κι αν ξέρετε τι κάνετε. Παγιώστε τις ρυθμίσεις με κλικ στο Save. Προκειμένου να επαληθεύσετε ότι όλα έχουν γίνει σωστά μέχρι στιγμής, πηγαίνετε στο Status → Interfaces.*

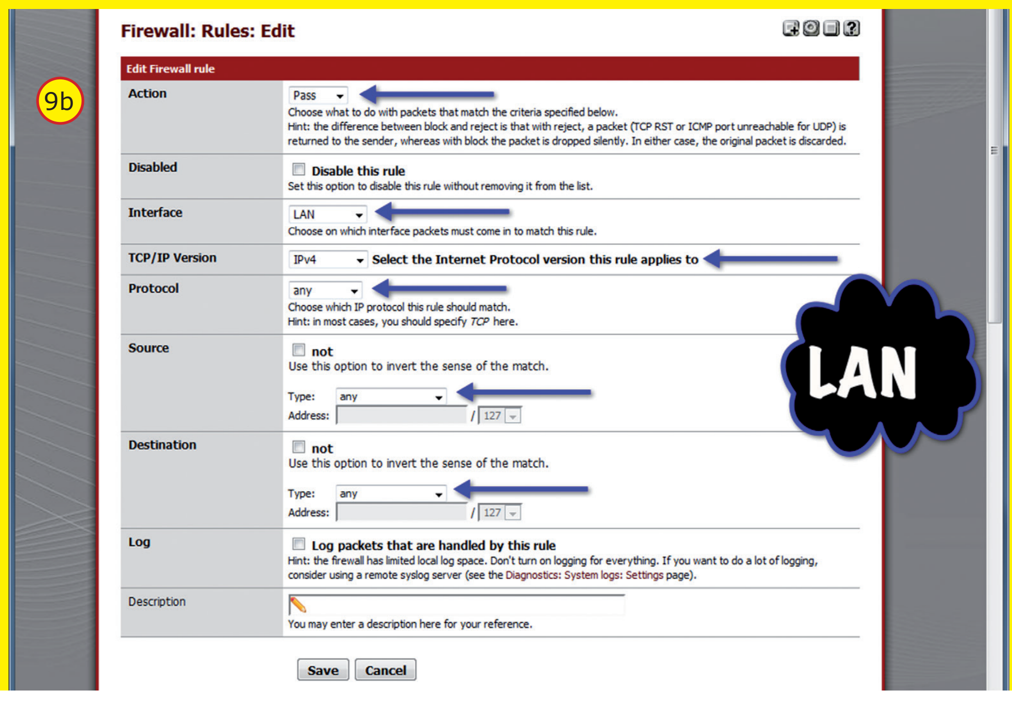


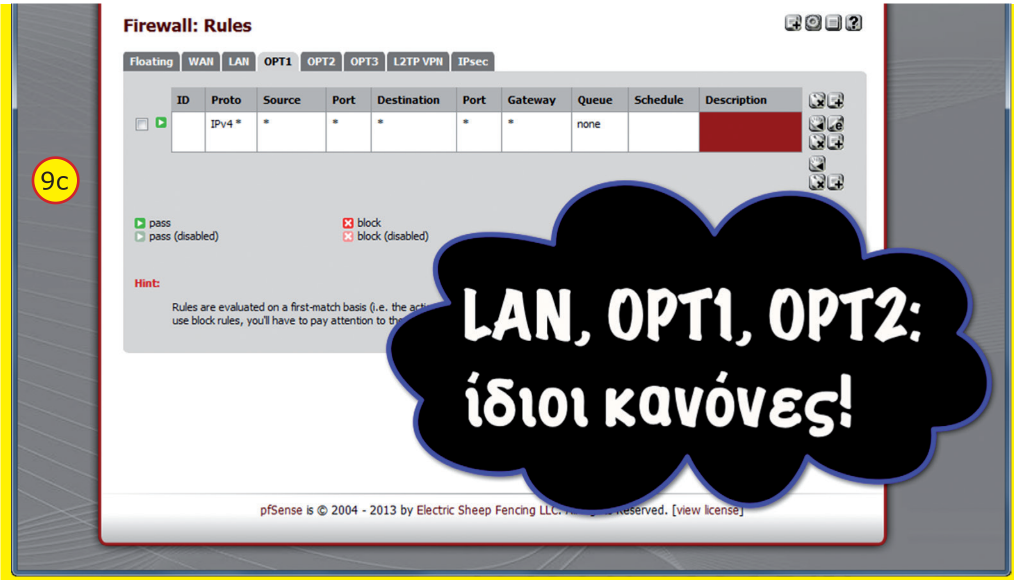
Βήμα 7. *Επιλέγουμε Services → DHCP Server και μεταβαίνουμε στην καρτέλα OPT3. Τσεκάρουμε το Enable DHCP server on OPT3 interface και στο Range δίνουμε ένα επιθυμητό εύρος διευθύνσεων IP, οι οποίες θα μοιράζονται αυτόματα στους πελάτες. Εμείς, π.χ., ορίσαμε το εύρος 192.168.80.100 - 192.168.80.150. Στο κάτω μέρος της σελίδας, πατάμε στο κουμπί Save.*



Βήμα 8. *Δίνουμε Services → DNS forwarder (υποθέτουμε ότι τον χρησιμοποιείτε κι εσείς). Στο πλαίσιο της κατηγορίας Interfaces, προς τα κάτω στη σελίδα, φροντίζουμε ώστε να είναι επιλεγμένο το OPT3 (εμάς, π.χ., ήταν το All) και τσεκάρουμε την επιλογή Strict Interface Binding. Επικυρώνουμε με κλικ στο Save.*


9a
Firewall: Rules: Edit
Edit Firewall rule
Action
Pass
Choose what to do with packets that match the criteria specified below.
Hint: the difference between block and reject is that with reject, a packet (TCP RST or ICMP port unreachable for UDP) is returned to the sender, whereas with block the packet is dropped silently. In either case, the original packet is discarded.
Disabled
Disable this rule
Set this option to disable this rule without removing it from the list.
Interface
OPT3
Choose on which interface packets must come in to match this rule.
TCP/IP Version
IPv4
Select the Internet Protocol version this rule applies to
Protocol
any
Choose which IP protocol this rule should match.
Hint: in most cases, you should specify TCP here.
Source
not
Use this option to invert the sense of the match.
Type: OPT3 subnet
Address: / 127
Destination
not
Use this option to invert the sense of the match.
Type: any
Address: / 127
Log
Log packets that are handled by this rule
Hint: the firewall has limited local log space. Don't turn on logging for everything. If you want to do a lot of logging, consider using a remote syslog server (see the Diagnostics: System logs: Settings page).
Description
Default "allow OPT3 to any" rule
You may enter a description here for your reference.
Save
Cancel
OPT3


9b
Firewall: Rules: Edit
Edit Firewall rule
Action
Pass
Choose what to do with packets that match the criteria specified below.
Hint: the difference between block and reject is that with reject, a packet (TCP RST or ICMP port unreachable for UDP) is returned to the sender, whereas with block the packet is dropped silently. In either case, the original packet is discarded.
Disabled
Disable this rule
Set this option to disable this rule without removing it from the list.
Interface
LAN
Choose on which interface packets must come in to match this rule.
TCP/IP Version
IPv4
Select the Internet Protocol version this rule applies to
Protocol
any
Choose which IP protocol this rule should match.
Hint: in most cases, you should specify TCP here.
Source
not
Use this option to invert the sense of the match.
Type: any
Address: / 127
Destination
not
Use this option to invert the sense of the match.
Type: any
Address: / 127
Log
Log packets that are handled by this rule
Hint: the firewall has limited local log space. Don't turn on logging for everything. If you want to do a lot of logging, consider using a remote syslog server (see the Diagnostics: System logs: Settings page).
Description
You may enter a description here for your reference.
Save
Cancel
LAN



Βήμα 9. *Μεταβαίνουμε στους κανόνες του firewall που αφορούν στο LAN interface, δίνοντας Firewall → Rules → LAN. Παρατηρήστε έναν κανόνα που έχει προσθέσει αυτόματα το pfSense: Τα πακέτα IPv4 που ξεκινούν απ' όλα τα μηχανήματα τα οποία είναι συνδεδεμένα στο LAN port, περνάνε ελεύθερα όπου κι αν πηγαίνουν. Θέλουμε έναν παρόμοιο κανόνα για το OPT3. Προκειμένου να προσθέσουμε νέο κανόνα για συγκεκριμένο network interface, πηγαίνουμε στην αντίστοιχη καρτέλα —εν προκειμένω στην OPT3— και κάνουμε κλικ στο κουμπάκι [+]. Δείτε το screenshot για το OPT3 και προσθέστε κι εσείς έναν ίδιο κανόνα. Αλλά δεν τελειώσαμε. Πηγαίνουμε τώρα ξανά στην καρτέλα με τους κανόνες του LAN. Δίπλα απ' αυτόν που μόλις παρατηρήσαμε, πριν φτιάξουμε τον κανόνα για το OPT3, κάνουμε κλικ στο [e]. Μπορούμε τώρα να τον τροποποιήσουμε και το μόνο που αλλάζουμε είναι την τιμή του Source, ώστε από LAN subnet να γίνει any. Επικυρώνουμε την αλλαγή και προσθέτουμε από έναν ίδιο ακριβώς κανόνα στα OPT1 και OPT2.*

Σημείωση. *Έτσι όπως φτιάξαμε τους κανόνες για τα LAN, OPT1 και OPT2, κάθε πελάτης του router είναι δυνατόν να επικοινωνεί με οποιονδήποτε άλλον πελάτη, ανεξαρτήτως σε πιο port είναι συνδεδεμένος (απευθείας ή μέσω switch). Αν επιθυμούσαμε να μην είναι εξ ορισμού δυνατή η επικοινωνία των μηχανημάτων που καταλήγουν σε διαφορετικά ports του router, τότε θα αφήναμε τον κανόνα για το LAN ως είχε και θα τον επαναλαμβάναμε για τα OPT1 και OPT2.*

Βήμα 10. *Αν όλα έχουν πάει καλά, τότε όλοι οι πελάτες του pfSense, σε όποιο port κι αν συνδέονται εκτός από εκείνο του WAN, θα παίρνουν διεύθυνση της μορφής 192.168.80.x. Ο router τους θα είναι ο 192.168.80.1, το ίδιο κι ο nameserver τους. Φυσικά, θα μπορούν να επικοινωνούν μεταξύ τους, ενώ θα βγαίνουν και στο Internet. Κάντε τις δοκιμές σας με ένα ή περισσότερα VMs πίσω από το pfSense, τα οποία θα έχουν δικτύωση τύπου host-only — αλλά για κάποιο από τα τρία host-only networks που ορίζουν οι αντίστοιχοι adapters του VirtualBox. (Αυτό το τελευταίο είναι για να εξομοιώσουμε την περίπτωση φυσικού router, όπου οι πελάτες του συνδέονται σε καθένα από τα Ethernet ports που ανήκουν στο ίδιο bridge group.)*

Παρατηρήστε, τέλος, ότι ενώ στα OPT1 και OPT2 interfaces δεν αποδώσαμε διευθύνσεις IP, στο LAN interface του pfSense αφήσαμε τη διεύθυνση IP που ήδη είχε. Αυτό το κάναμε για λόγους εύκολης διαχείρισης: Μέσω του LAN port είμασταν συνδεδεμένοι στο web panel του pfSense, και δουλεύαμε. Ωστόσο, γενικά κάτι τέτοιο δεν είναι απαραίτητο. Κάλλιστα, δηλαδή, ένα bridge interface μπορεί να γεφυρώνει (ενεργά) network interfaces που στερούνται διευθύνσεων IP. Ένα τέτοιο σενάριο θα δούμε και στη συνέχεια, όπου δουλεύουμε με το OpenBSD.

Bridging με το OpenBSD

Μια εναλλακτική του pfSense πρόταση για τον DIY router σας, αποτελεί το OpenBSD. Δεν θα ήταν υπερβολή αν λέγαμε ότι το εν λόγω λειτουργικό είναι συνώνυμο της ασφάλειας — και σίγουρα είναι πολύ, μα πάρα πολύ, ελαφρύτερο από το pfSense. Αν έως σήμερα δεν είχατε την ευκαιρία να παίξετε με το OpenBSD, γνωρίζοντας τι θ' ανακαλύψετε σχεδόν σας ζηλεύουμε! Ξεκινήστε την ενασχόλησή σας με το 10ο επεισόδιο της 1ης σεζόν του deltaCast:

`http://deltahacker.gr/deltacast-s01e10`

Η αναλυτική επίδειξη της δημιουργίας bridge στο OpenBSD που ακολουθεί, γίνεται στο πλαίσιο εικονικού δικτύου εντός του οποίου έχουμε ήδη ένα VirtualBox VM με OpenBSD, σε ρόλο router (βλ. το deltaCast s01e10). Το εν λόγω VM έχει τέσσερις network adapters.

Adapter 1. Είναι τύπου bridged και γεφυρωμένος από το VirtualBox με την αληθινή κάρτα δικτύου του host computer — συγκεκριμένα μ' αυτή που συνδέεται στο αληθινό modem/router του τοπικού δικτύου. Ο Adaper 1 έχει το ρόλο του WAN interface και για το OpenBSD είναι το em0.

Adapter 2, Adapter 3 & **Adapter 4.** Όλοι τους είναι τύπου host-only, ανήκουν σε τρία διαφορετικά host-only networks του VirtualBox και είναι σε Promiscuous Mode. Έχουν και τα τρία ρόλο LAN adapter και για το OpenBSD είναι τα em1, em2 και em3 αντίστοιχα.

Το bridge interface που θα δημιουργήσουμε σε πολύ λίγο θα γεφυρώνει τα em1, em2 και em3. Αυτή τη φορά κανένα εκ των em1, em2, em3 δεν θα 'χει διεύθυνση IP: Τα μηχανήματα πίσω από τον router θα βλέπουν μόνο τη διεύθυνση του bridge interface. Ακριβώς γι' αυτό δουλεύουμε τοπικά, από την κονσόλα του OpenBSD VM και φυσικά μέσα από τον λογαριασμό του root.

Βήμα 1. Ρυθμίζουμε τα em1, em2, em3 ώστε να είναι ενεργά — αλλά χωρίς να έχουν διευθύνσεις IP:

```
# echo up > /etc/hostname.em1
# echo up > /etc/hostname.em2
# echo up > /etc/hostname.em3
```

Ρυθμίζουμε το network interface της γέφυρας (vether0), δίνοντάς του μια κατάλληλη, non-routable διεύθυνση IP. Εμείς, π.χ., επιλέξαμε να του αντιστοιχίσουμε τη διεύθυνση 192.168.90.1, αφού ούτε το αληθινό αλλά ούτε και κάποιο από τα host-only networks του VirtualBox δεν ήταν το 192.168.90.0/24.

```
# echo "inet 192.168.90.1 255.255.255.0" > /etc/hostname.vether0
```

Για τη ρύθμιση της ίδιας της γέφυρας δημιουργούμε, π.χ., με το nano, το νέο αρχείο ονόματι /etc/hostname.bridge0:

```
# nano /etc/hostname.bridge0
```

Του προσθέτουμε τις ακόλουθες γραμμές:

```
add vether0
add em1
```

```
add em2
add em3
blocknonip em1
blocknonip em2
blocknonip em3
up
```

Τις αλλαγές στο αρχείο τις γράφουμε με [CTRL+O] και [Enter], ενώ το nano το εγκαταλείπουμε με [CTRL+X]. Πριν ενεργοποιήσουμε τη γέφυρα, ας κάνουμε και τις απαραίτητες τροποποιήσεις στις ρυθμίσεις που αφορούν στον DHCP server, καθώς και στο pf. Ανοίγουμε κατ' αρχάς το αρχείο /etc/rc.conf.local:

```
# nano /etc/rc.conf.local
```

Αν έχουμε ακολουθήσει τις οδηγίες του deltaCast s01e10, θα δούμε την ακόλουθη γραμμή:

```
dhcpd_flags="em0"
```

Την αλλάζουμε ώστε να είναι ακριβώς όπως η ακόλουθη:

```
dhcpd_flags="vether0"
```

Οφείλουμε βεβαίως να φροντίσουμε και τον ίδιο τον DHCP server. Δείτε το δικό μας αρχείο ρυθμίσεων και τροποποιήστε αναλόγως το δικό σας (μιλάμε για το /etc/dhcpd.conf):

```
option  domain-name "parabing.net";
option  domain-name-servers 8.8.8.8, 8.8.4.4;

subnet 192.168.90.0 netmask 255.255.255.0 {
    option routers 192.168.90.1;
    range 192.168.90.100 192.168.90.150;
}
```

Μένει, τέλος, να κάνουμε τις απαραίτητες τροποποιήσεις και στο ήδη υπάρχον /etc/pf.conf. Ιδού το δικό μας, μετά από όλες τις αλλαγές:

```
wan_if="em0"
int_if="vether0 em1 em2 em3"

allowed_icmp_types="echoreq"
allowed_tcp_service_ports="{ 22 }"

set block-policy drop
set loginterface $wan_if
set skip on lo
```

```
match out on $wan_if inet from !($wan_if:network) to any nat-to ($wan_
if:0)

block in log
pass out quick

antispoof quick for { lo $wan_if }

pass in on $wan_if inet proto tcp from any to ($wan_if) port $allowed_
tcp_service_ports

pass in inet proto icmp all icmp-type $allowed_icmp_types

pass in on { $int_if }
```

(Μπορείτε αν θέλετε να το κατεβάσετε από το http://bit.ly/dh027pfconf.) Ο γρηγορότερος τρόπος για να εφαρμόσουμε *όλες* τις αλλαγές είναι με μια επανεκκίνηση:

```
# reboot
```

Αυτό ήταν. Αν όλα έχουν πάει καλά, κάθε VM-πελάτης του OpenBSD, ανεξαρτήτως του (εικονικού) port στο οποίο είναι συνδεδεμένος, θα έχει διεύθυνση IP από το εύρος 192.168.90.100-192.168.90.150, ενώ για nameservers θα χρησιμοποιεί τους 8.8.8.8, 8.8.4.4 (της Google). Όλοι οι πελάτες θα επικοινωνούν μεταξύ τους, θα βγαίνουν στο Internet μέσω του OpenBSD VM και θα προστατεύονται από το pf.

Και πριν σας αφήσουμε να παρακολουθήσετε μια επίδειξη επίθεσης MiTM που στηρίζεται στο wireless bridging (βλ. άρθρο σελ. 64), μία μόνο ερώτηση θα σας κάνουμε: Έχετε αποφασίσει πιο Soekris board θ' αγοράσετε ή μήπως το μελετάτε ακόμα; :P

1

Το **deltaHacker** είναι το ένα και **μοναδικό** περιοδικό με θεματολογία το ethical **hacking**, τα **δίκτυα** και την **ασφάλεια**, τον **προγραμματισμό** και τα **ηλεκτρονικά**.

Το **deltaHacker** δεν γυρνάει εδώ κι εκεί στα περίπτερα, αλλά για δεύτερη χρονιά στη σειρά **έρχεται κατευθείαν σ' εσάς**.

Αν είστε από εκείνους που δεν βλέπουν τον υπολογιστή τους ως άλλη μια οικιακή συσκευή και δεν αντιμετωπίζετε το Internet ως ένα δίκτυο που αρχίζει και τελειώνει με το Facebook και το Twitter, τότε είναι **109% βέβαιο ότι θα αγαπήσετε το deltaHacker**, καθώς κι ότι θα θέλετε την παρέα του **κάθε μήνα**.

Παραγγείλετε τώρα μία από τις συμφέρουσες συνδρομές από το

http://deltahacker.gr/pdf12order
(ηλεκτρονική έκδοση)

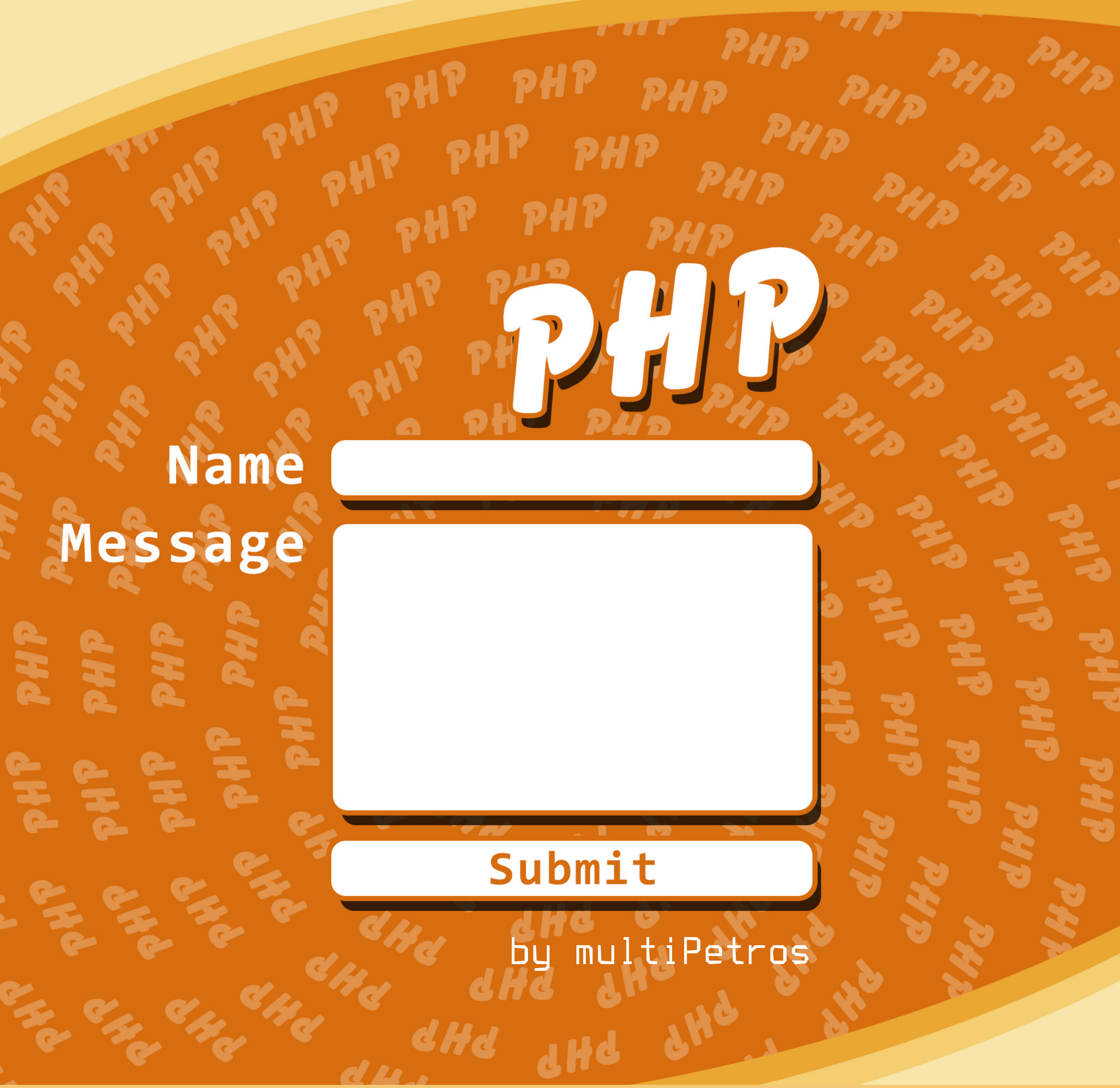
ή από το

http://deltahacker.gr/order
(έντυπη έκδοση, έως το τεύχος 019)

Σημείωση: Την ύλη **όλων των τευχών** που έχουν κυκλοφορήσει ως τώρα μπορείτε να τη βλέπετε στο

deltahacker.gr/category/magazine-issues

Skill: Intermediate
Tags: LAMP, PHP, fopen, fclose, fwrite, fread, frewind, fseek, feof



Web development σε περιβάλλον LAMP [μέρος 4]

Έχοντας εξετάσει και την αντικειμενοστρεφή πλευρά της PHP, ξεμπερδέψαμε με ένα μεγάλο μέρος της θεωρητικής γνωριμίας με τη γλώσσα. Και μιλάμε για θεωρητική γνωριμία, διότι η ουσιαστική γνώση προϋποθέτει και την πρακτική ενασχόληση. Σε αυτό το μέρος της σειράς μας θα αρχίσουμε να αποκτάμε πιο στέρεη γνώση, χτίζοντας (επιτέλους) την πρώτη μας δικτυακή εφαρμογή!

Πριν ξεκινήσουμε πρέπει να ξεκαθαρίσουμε τι εννοούμε όταν λέμε ότι θα φτιάξουμε μια δικτυακή εφαρμογή. Είναι γεγονός ότι με τις γνώσεις που έχουμε αποκτήσει ως τώρα μπορούμε να γράψουμε αρκετά μεγάλα, σύνθετα και εντυπωσιακά προγράμματα. Η διαδικασία εκμάθησης, όμως, απαιτεί σύνεση. Στο επίπεδο που βρισκόμαστε, η μελέτη μιας τεράστιας εφαρμογής δεν θα προσέφερε τίποτα ουσιαστικό. Αν για παράδειγμα εξετάζαμε τη δομή και τη λειτουργία ενός ολόκληρου CMS, θα σας φλομώναμε στη θεωρία και, στο τέλος, όχι μόνο δεν θα είχατε μάθει κάτι χρήσιμο, αλλά θα είχατε κουραστεί κι ενδεχομένως απογοητευτεί. Αυτό που πρέπει να μελετήσουμε είναι μια μικρή εφαρμογή, η οποία θα αλληλεπιδρά με το χρήστη, θα μπορεί να αποθηκεύει δεδομένα για μελλοντική χρήση και, τέλος πάντων, θα πραγματοποιεί μια κάποια εργασία. Η εφαρμογή που επιλέξαμε να παρουσιάσουμε είναι ένα βιβλίο επισκεπτών (GuestBook), το οποίο θα επιτρέπει στους χρήστες να καταχωρούν τα μηνύματά τους, ενώ θα προβάλλει και τα ήδη αποθηκευμένα.

Βασικός χειρισμός αρχείων

Για την αποθήκευση, την ανάκτηση και γενικότερα τη διαχείριση δεδομένων, οι βάσεις δεδομένων αποτελούν την καλύτερη δυνατή επιλογή. Δεν σκοπεύουμε να ασχοληθούμε σ' αυτό το άρθρο με την MySQL, τη βάση δεδομένων του πακέτου LAMP. Θα προτιμήσουμε την παλιά, απλοϊκή και σίγουρη λύση της αποθήκευσης σε αρχεία. Εξάλλου, παρά τις ευκολίες που προσφέρουν οι βάσεις δεδομένων, η χρησιμότητα των αρχείων παραμένει μεγάλη. Επομένως, μαθαίνοντας PHP έχει αξία να γνωρίσουμε όλες τις λειτουργίες που σχετίζονται με το χειρισμό των αρχείων.

Για την ανάγνωση ενός αρχείου ή την εγγραφή δεδομένων σ' αυτό, πρέπει πρώτα να το ανοίξουμε. Αυτό γίνεται με τη συνάρτηση fopen, στην οποία δηλώνουμε τη διαδρομή του αρχείου και την κατάσταση στην οποία θέλουμε να το θέσουμε. Ως κατάσταση νοείται ο τρόπος με τον οποίο θέλουμε να εργαστούμε (ανάγνωση, εγγραφή, ανάγνωση κι εγγραφή κ.λπ.). Ας δούμε τη σύνταξη της fopen:

```
$fp = fopen('path/to/file', 'r+') ;
```

Όταν εκτελείται η συνάρτηση fopen, επιστρέφει έναν πόρο (resource). Με απλά λόγια, επιστρέφει έναν δείκτη, μέσω του οποίου μπορούμε να χειριστούμε το αρχείο. Η δεύτερη παράμετρος δηλώνει τον τρόπο εργασίας επί του αρχείου και μπορεί να πάρει μόνο μία από τις τιμές που περιλαμβάνει ο ακόλουθος πίνακας:

r	Ανάγνωση
r+	Ανάγνωση και εγγραφή
w	Εγγραφή. Αν υπάρχει το αρχείο, τα περιεχόμενά του θα επικαλυφθούν.
w+	Ανάγνωση και εγγραφή. Αν υπάρχει το αρχείο, τα περιεχόμενά του θα επικαλυφθούν.
x	Ασφαλής εγγραφή. Αν υπάρχει το αρχείο, δεν θα ανοιχτεί.
x+	Ανάγνωση και εγγραφή και ανάγνωση. Αν υπάρχει το αρχείο, δεν θα ανοιχτεί.
a	Εγγραφή με προσάρτηση δεδομένων στο τέλος του αρχείου (append).
a+	Ανάγνωση και εγγραφή, με προσάρτηση των δεδομένων στο τέλος του αρχείου (append).

Πριν προχωρήσουμε, σημειώστε ότι αφού ανοίξουμε ένα αρχείο και όταν ολοκληρώσουμε την εργασία μας, πρέπει και να το κλείσουμε. Η συνάρτηση που χρησιμοποι-

ούμε για το κλείσιμο ενός ανοιχτού αρχείου είναι η fclose. Σε αυτήν δίνουμε σαν παράμετρο τον πόρο που είχαμε λάβει από την fopen. Η σύνταξή της είναι απλούστατη:

```
fclose($fp) ;
```

Η συνάρτηση επιστρέφει true αν το κλείσιμο του αρχείου ολοκληρώθηκε με επιτυχία, ή false αν προέκυψε κάποιο σφάλμα.

Εγγραφή δεδομένων

Για να γράψουμε σ' ένα αρχείο χρειαζόμαστε κατ' αρχάς τον αντίστοιχο πόρο, τον οποίο λαμβάνουμε εκτελώντας την fopen. Στη συνέχεια, αρκεί να καλέσουμε τη συνάρτηση fwrite. Αυτή δέχεται ως παραμέτρους τον πόρο του αρχείου, καθώς και μια μεταβλητή τύπου string με τα προς εγγραφή δεδομένα. Προαιρετικά μπορούμε να δώσουμε και μια τρίτη παράμετρο, που δηλώνει το μήκος του string με τα δεδομένα[1]:

```
fwrite($fp, $string [,$length]) ;
```

Η συνάρτηση fwrite επιστρέφει έναν ακέραιο αριθμό, που αναπαριστά το πλήθος των bytes που γράφτηκαν στο αρχείο. Στην περίπτωση που η εγγραφή αποτύχει, η fwrite επιστρέφει την τιμή False.

Αντίστοιχη με την fwrite (αν και πιο εύχρηστη) είναι η file_put_contents. Αυτή η συνάρτηση αυτοματοποιεί όλη τη διαδικασία και συγκεκριμένα εκτελεί τις ακόλουθες εργασίες: Ανοίγει το αρχείο που δηλώνουμε, γράφει τα επιθυμητά δεδομένα και στη συνέχεια κλείνει το αρχείο. Η file_put_contents δέχεται ως παραμέτρους τη διαδρομή προς το αρχείο και τα δεδομένα που θέλουμε να γράψουμε (είτε στη μορφή μιας μεταβλητής string είτε σε έναν πίνακα). Προαιρετικά μπορούμε να δώσουμε και κάποια flags, που ενεργοποιούν πρόσθετες λειτουργίες. Παρακάτω φαίνεται η σύνταξη της συνάρτησης καθώς κι ένας πίνακας με τα διαθέσιμα flags:

```
file_put_contents($filepath, $string [,flags]) ;
```

FILE_USE_INCLUDE_PATH	Αναζητά το αρχείο στους καταλόγους που περιλαμβάνει το δοσμένο path, μέσα στο αρχείο ρυθμίσεων php.ini.
FILE_APPEND	Αν το αρχείο υπάρχει ήδη, προσαρτά τα δεδομένα στο τέλος αυτού και δεν τα επικαλύπτει.
LOCK_EX	Κλειδώνει το αρχείο για αποκλειστική χρήση.

Ανάγνωση δεδομένων

Για να διαβάσουμε τα περιεχόμενα ενός αρχείου, έχουμε περισσότερες επιλογές. Η πιο απλή μέθοδος είναι να χρησιμοποιήσουμε τη συνάρτηση file_get_contents. Η εν λόγω συνάρτηση, σε αντιστοιχία με την file_put_contents, ανοίγει το αρχείο, διαβάζει τα δεδομένα και το κλείνει αυτόματα. Η μόνη παράμετρος που απαιτεί η συνάρτηση file_get_contents είναι η διαδρομή και το όνομα του επιθυμητού αρχείου:

```
file_get_contents($filepath) ;
```

Όταν ολοκληρωθεί η ανάγνωση, η συνάρτηση επιστρέφει ένα string με τα περιεχόμενα του αρχείου. Μια παρόμοια συνάρτηση είναι η file, που επιστρέφει τα δεδομένα του δοσμένου αρχείου σε έναν πίνακα. Συγκεκριμένα, κάθε στοιχείο αυτού του

1. Αν το μήκος που δώσουμε είναι μεγαλύτερο από το μήκος της μεταβλητής με τα δεδομένα, η fwrite θα σταματήσει όταν φτάσει στο τέλος της μεταβλητής.

πίνακα είναι μια μεταβλητή string και περιλαμβάνει το δεδομένα μιας γραμμής του αρχείου:

```
file($filepath) ;
```

Υπάρχουν περιπτώσεις κατά τις οποίες δεν θέλουμε να διαβάσουμε ένα ολόκληρο αρχείο (π.χ., γιατί είναι πολύ μεγάλο). Τότε βολεύει να διαβάσουμε τα περιεχόμενα του αρχείου γραμμή προς γραμμή. Κάτι τέτοιο επιτυγχάνεται με τη συνάρτηση fgets, στην οποία δίνουμε σαν παράμετρο το επιθυμητό αρχείο. Προαιρετικά, μπορούμε να δηλώσουμε και το πλήθος των χαρακτήρων που θέλουμε να διαβάσει από την εκάστοτε γραμμή. Πάντως, αν δεν ορίσουμε την προαιρετική παράμετρο, η fgets επιστρέφει ολόκληρη την τρέχουσα γραμμή:

```
$line_string = fgets($fp [,length]) ;
```

Τώρα, βέβαια, θα αναρωτιέστε πώς μπορούμε να διαβάσουμε τη δεύτερη γραμμή του αρχείου, την τρίτη και πάει λέγοντας. Για να διαβάσουμε την επόμενη γραμμή, αρκεί να ξανακαλέσουμε τη συνάρτηση fgets! Για να μην προκύψει όμως κάποιο σφάλμα, θα πρέπει με κάποιον τρόπο να γνωρίζουμε ότι δεν έχουμε φτάσει στο τέλος του αρχείου. Αυτό μπορούμε να το μάθουμε από τη συνάρτηση feof. Η εν λόγω συνάρτηση δέχεται ως παράμετρο έναν πόρο αρχείου και επιστρέφει true αν ο δείκτης θέσης του αρχείου έχει φτάσει στο τέλος. Το παρακάτω απόσπασμα κώδικα εμφανίζει τα περιεχόμενα ενός αρχείου γραμμή προς γραμμή, φροντίζοντας να τις απαριθμήσει:

```
$i = 0 ;
while(!feof($fp)){
    $line = fgets($fp) ;
    $i++ ;
echo "Line $i : $line\n" ;
}
```

Μια πιο εξεζητημένη παραλλαγή της fgets αποτελεί η fgetss. Αυτή η συνάρτηση επιστρέφει τα περιεχόμενα του αρχείου γραμμή προς γραμμή, αφού αφαιρέσει όλες τις ετικέτες (tags) HTML, καθώς και τυχόν χαρακτήρες NULL. Η fgets δέχεται σαν παράμετρο το όνομα ενός αρχείου και δύο προαιρετικές παραμέτρους. Η πρώτη ορίζει το πλήθος των χαρακτήρων που θα επιστραφούν, ενώ η δεύτερη δηλώνει τις ετικέτες (tags) που θα αγνοηθούν και *δεν* θα αφαιρεθούν.

Μερικές φορές, η φύση της εργασίας μας καθιστά την ανάγνωση κατά γραμμή εντελώς ασύμφορη και μπελαλίδικη. Σε αυτές τις περιπτώσεις, βολεύει να διαβάζουμε το αρχείο χαρακτήρα προς χαρακτήρα. Κάτι τέτοιο γίνεται εύκολα με τη συνάρτηση fgetc, η οποία δέχεται σαν παράμετρο έναν πόρο αρχείου και επιστρέφει έναν μόλις χαρακτήρα. Αν την καλέσουμε εκ νέου, επιστρέφει τον αμέσως επόμενο χαρακτήρα και αυξάνει κατά ένα byte τον «δείκτη θέσης». Για να ελέγξουμε αν έχουμε φτάσει στο τέλος του αρχείου, μπορούμε και πάλι να χρησιμοποιήσουμε τη συνάρτηση feof. Το απόσπασμα που ακολουθεί διαβάζει κάποιο αρχείο χαρακτήρα προς χαρακτήρα, αντικαθιστά τις τελείες με κόμματα και το αντίστροφο:

```
while(!feof($fp)){
    $char = fgetc($fp) ;
    if ($char == '.')
```

```
echo ',' ;
elseif ($char == ',')
echo '.' ;
else
echo $char ;
}
```

Για το τέλος αφήσαμε τη συνάρτηση fread, η οποία επιτρέπει να διαβάζουμε όσους χαρακτήρες επιθυμούμε, ξεκινώντας από την τρέχουσα θέση (αυτήν που υπαγορεύει ο «δείκτης θέσης»). Νομίζουμε ότι οι δύο παράμετροι που δέχεται η συνάρτηση είναι αυτονόητες:

```
fread($fp, $length) ;
```

Η συγκεκριμένη συνάρτηση συνδυάζεται συνήθως με τις rewind και fseek. Ας ρίξουμε μια ματιά και σ' αυτές:

```
rewind($fp) ;
```

Η rewind δέχεται ως παράμετρο έναν πόρο αρχείου και μηδενίζει τον αντίστοιχο «δείκτη θέσης», ώστε η επόμενη εργασία ανάγνωσης να ξεκινήσει από την αρχή.

```
fseek($fp, $offset [,$whence]) ;
```

Η fseek δέχεται ως παράμετρο τον πόρο ενός αρχείου και αυξάνει τον «δείκτη της θέσης» κατά τόσα bytes, όσα δηλώνει η παράμετρος offset. Η τελευταία παράμετρος είναι προαιρετική και μπορεί να δεχτεί μόνο μία από τις ακόλουθες τρεις τιμές: SEEK_CUR, SEEK_SET και SEEK_END. Στην πρώτη περίπτωση η συνάρτηση παρουσιάζει τη καθιερωμένη συμπεριφορά (αυτήν που περιγράψαμε παραπάνω), ενώ στη δεύτερη περίπτωση, ο «δείκτης θέσης» παίρνει την τιμή της παραμέτρου offset. Στην τρίτη περίπτωση, ο «δείκτης θέσης» παραπέμπει σε μια θέση του αρχείου, που απέχει τόσα bytes από το τέλος, όσα δηλώνει η παράμετρος offset. Αυτή η λειτουργία μπορεί να μοιάζει παράξενη, αλλά όταν θα σας χρειαστεί (και να είστε σίγουροι ότι κάποια στιγμή θα σας χρειαστεί) θα την θυμηθείτε και θα χαρείτε για την ύπαρξή της.

Φάκελοι και λοιποί έλεγχοι

Δουλεύοντας με αρχεία, είναι βέβαιο ότι κάποια στιγμή θα χρειαστεί να ελέγξουμε για την ύπαρξη κάποιου. Σε αυτή τη δουλειά βοηθά η συνάρτηση file_exists, που δέχεται σαν μοναδική παράμετρο τη θέση και το όνομα ενός αρχείου. Αν το αρχείο υπάρχει, η συνάρτηση επιστρέφει true, διαφορετικά επιστρέφει false. Ας δούμε κι ένα απλό παράδειγμα:

```
echo "Το αρχείο $filepath " ;
echo file_exists($filepath)? "υπάρχει" : "δεν υπάρχει" ;
```

Για να μάθουμε το μέγεθος ενός αρχείου σε bytes, μπορούμε να καλέσουμε τη συνάρτηση filesize. Το παρακάτω παράδειγμα χρήσης μιλάει από μόνο του:

```
echo "Το αρχείο $filepath , έχει μέγεθος" .filesize($filepath) ;
```

Ορισμένες φορές θα χρειαστεί να αντιγράψουμε, να διαγράψουμε ή απλά να μετονομάσουμε ένα αρχείο. Η PHP προσφέρει έτοιμες συναρτήσεις και για αυτές τις

εργασίες. Πρόκειται για τις copy, unlink και rename, αντίστοιχα:

```
copy($source, $destination)
unlink($filepath) ;
rename($origin, $target) ;
```

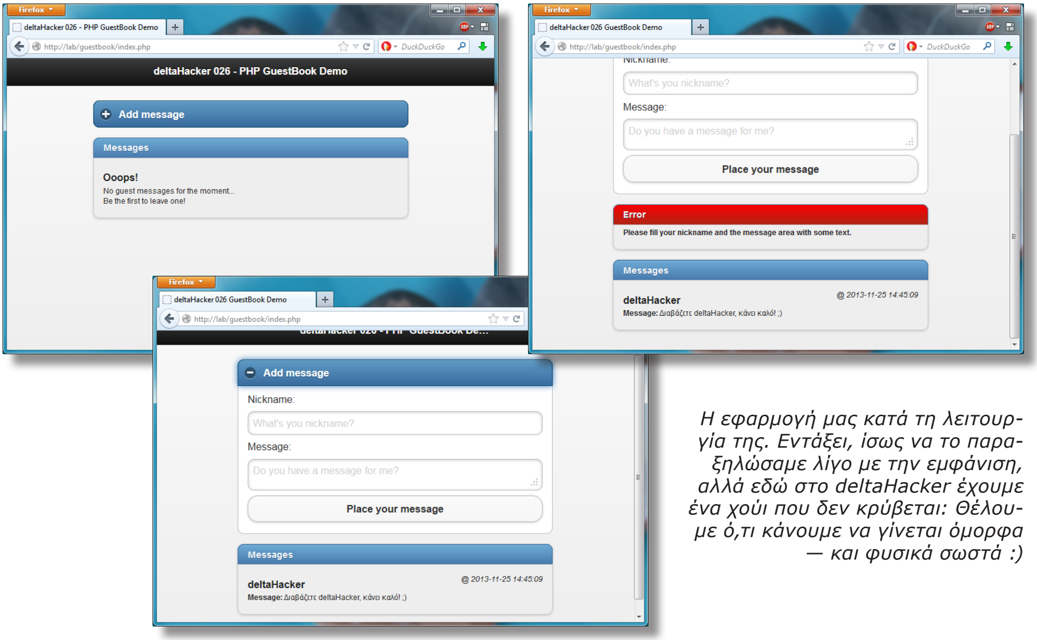
Η ερμηνεία των παραμέτρων που δέχεται κάθε συνάρτηση είναι μάλλον προφανής και δεν χρειάζεται να πούμε περισσότερα. Θα πρέπει όμως να σημειώσουμε μια κρίσιμη λεπτομέρεια για τη συμπεριφορά της copy: Όταν το αρχείο προορισμού υπάρχει ήδη, η συνάρτηση αντιγραφής το επικαλύπτει χωρίς να παραπονεθεί και χωρίς να μας ειδοποιήσει! Γι' αυτό το λόγο, πριν κάνουμε μια αντιγραφή είναι φρόνιμο να χρησιμοποιούμε και τη συνάρτηση file_exists και να τσεκάρουμε αν υπάρχει ήδη το αρχείο προορισμού.

Τέλος, η PHP περιλαμβάνει μία συνάρτηση για τη δημιουργία καταλόγων, καθώς και μία για τη διαγραφή καταλόγων:

```
mdkir($dirname) ;
rmdir($dirname) ;
```

Η πρώτη μας εφαρμογή!

Όπως είπαμε αρχικά, η εφαρμογή που θα εξετάσουμε είναι ένα απλοϊκό «βιβλίο επισκεπτών». Όσοι είχατε ασχοληθεί με τη δημιουργία ιστοσελίδων στο μακρινό παρελθόν (πριν από την έλευση του WEB 2.0), ενδέχεται να θυμόσαστε ότι οι εταιρίες web hosting παρείχαν διάφορες εφαρμογές στους χρήστες, με τη μορφή cgi script. Μια δημοφιλής εφαρμογή του είδους ήταν το «βιβλίο επισκεπτών». Η συγκεκριμένη εφαρμογή επέτρεπε στους επισκέπτες μιας ιστοσελίδας να αφήνουν δημόσια μηνύματα προς τον ιδιοκτήτη, συμπληρώνοντας μια κατάλληλη φόρμα. Φυσικά, η ίδια



Η εφαρμογή μας κατά τη λειτουργία της. Εντάξει, ίσως να το παραξηλώσαμε λίγο με την εμφάνιση, αλλά εδώ στο deltaHacker έχουμε ένα χούι που δεν κρύβεται: Θέλουμε ό,τι κάνουμε να γίνεται όμορφα — και φυσικά σωστά :)

εφαρμογή φρόντιζε και για την προβολή όλων των αποθηκευμένων μηνυμάτων. Με τις γνώσεις που έχουμε αποκτήσει ως τώρα, μπορούμε εύκολα να φτιάξουμε ένα «βιβλίο επισκεπτών» στην PHP.

Η λειτουργία της εφαρμογής είναι απλή. Μέσω μιας φόρμας, ο επισκέπτης του site θα δίνει ένα nickname και θα εισάγει το επιθυμητό μήνυμα. Αυτά τα δεδομένα θα μεταφέρονται στον server κατά την υποβολή της φόρμας και από 'κει θα αποθηκεύονται σε ένα αρχείο απλού κειμένου (plain text file). Τα δεδομένα από τα πεδία της φόρμας (nickname και μήνυμα), όπως και η ημερομηνία που προστίθεται στο «πακέτο» αυτόματα, αποθηκεύονται σε μια μόνο γραμμή και χωρίζονται μεταξύ τους με τον χαρακτήρα TAB. Το πρόγραμμά μας θα διαβάζει το εν λόγω αρχείο γραμμή προς γραμμή, θα αναλύει τα δεδομένα και θα τα παρουσιάζει στον browser του επισκέπτη. Μπορείτε να κατεβάσετε τον πλήρη κώδικα της εφαρμογής μας από το http://bit.ly/dh027guestbooksrc. Στη συνέχεια θα εξετάσουμε τη λειτουργία του κώδικα, τμηματικά. Για να διευκολύνουμε τη μελέτη έχουμε αφαιρέσει τις περισσότερες ετικέτες HTML και τα αποσπάσματα που ακολουθούν περιέχουν σχεδόν αποκλειστικά τον κώδικα σε PHP.

```
07 define('FILE','../ext_storage/data') ;
08 $savedGood = true ;
09 $emptyForm = false ;
```

Ξεκινώντας ορίζουμε τη σταθερά FILE, η οποία περιέχει τη διαδρομή προς το αρχείο όπου θα διατηρούνται τα μηνύματα των επισκεπτών. Μια καλή πρακτική είναι η διαδρομή του αρχείου να βρίσκεται έξω από τη ρίζα του Document Root, έτσι ώστε τα περιεχόμενα του αρχείου να μην είναι άμεσα προσβάσιμα από τον ιστό. Για την εφαρμογή μας αυτό δεν είναι καθόλου σημαντικό, αλλά πρόκειται για μια καλή τακτική και οφείλαμε να την αναφέρουμε ;) Στις γραμμές 08 και 09 ορίζουμε δύο μεταβλητές, με τη βοήθεια των οποίων θα μπορούμε αργότερα να εντοπίζουμε όποια σφάλματα συμβαίνουν κατά τη λειτουργία του προγράμματος και να τα αντιμετωπίζουμε.

```
12 function clear_text($text){
13     return str_replace(array("\t","\n"), array(" "," "), $text) ;
14 }
```

Η συνάρτηση που ξεκινάει στη γραμμή 12 απαλείφει ορισμένους ανεπιθύμητους χαρακτήρες από το μήνυμα του χρήστη. Συγκεκριμένα, απαλείφει τον χαρακτήρα TAB και εκείνον της αλλαγής γραμμής. Αυτό είναι αναγκαίο, εξαιτίας του τρόπου αποθήκευσης των δεδομένων στο αρχείο. Η αφαίρεση των χαρακτήρων γίνεται με τη βοήθεια της συνάρτησης str_replace. Σ' αυτή δίνουμε έναν πίνακα με τους χαρακτήρες που θέλουμε να ξεφορτωθούμε, καθώς κι έναν πίνακα με τους χαρακτήρες που θα αντικαταστήσουν τους ανεπιθύμητους.

```
16 if(isset($_POST['nickname']) && isset($_POST['message'])){
17     $nickname = trim(clear_text(htmlspecialchars(strip_tags($_
       POST['nickname'])))) ;
18     $message = trim(clear_text(htmlspecialchars(strip_tags($_
       POST['message'])))) ;
19     if($nickname == "" || $message == ""){
20         $emptyForm = true ;
```

```
21     }else{
23         $data = date("Y-m-d H:i:s")."\t".$nickname."\t".$message."\n" ;
24         $savedGood = file_put_contents(FILE, $data, FILE_APPEND) ;
25     }
26 }
```

Στη γραμμή 16 ελέγχουμε αν ο πίνακας _POST περιλαμβάνει τα στοιχεία nickname και message. Αν ισχύει κάτι τέτοιο, ο κώδικας θεωρεί ότι έγινε υποβολή της φόρμας και ότι έχουν φτάσει τα δεδομένα ενός νέου μηνύματος. Στις γραμμές 17 και 18 εκτελούμε έναν καθαρισμό των εισερχόμενων δεδομένων μέσω μιας «τηλεσκοπικής», αν μας επιτρέπεται ο όρος, κλήσης τεσσάρων συναρτήσεων. Με αυτό εννοούμε ότι οι συναρτήσεις καλούνται αλυσιδωτά και η τιμή που επιστρέφει καθεμία δίνεται σαν παράμετρος στην επόμενη. Αντίθετα με το πώς διαβάζουμε τις γραμμές με τον κώδικα, η κλήση των συναρτήσεων ξεκινάει από δεξιά προς τ' αριστερά: Πρώτα η strip_tags, μετά η htmlspecialchars, στη συνέχεια η clear_text και τέλος η trim.

Στη γραμμή 19 ελέγχουμε αν ο χρήστης συμπλήρωσε όλα τα πεδία της φόρμας και, τέλος πάντων, το αν έχουν κάποια τιμή οι αντίστοιχες μεταβλητές. Αν κάποια από τις μεταβλητές δεν έχει τιμή, δίνουμε στη μεταβλητή emptyForm την τιμή true (γραμμή 20). Διαφορετικά, δημιουργούμε μια συμβολοσειρά που αποτελείται από την τρέχουσα ημερομηνία και ώρα, έναν χαρακτήρα TAB, την τιμή της μεταβλητής nickname, έναν ακόμα χαρακτήρα TAB, την τιμή της μεταβλητής message και, τέλος, έναν χαρακτήρα αλλαγής γραμμής. Όπως καταλαβαίνετε, ολόκληρη αυτή η συμβολοσειρά θα γραφτεί στο αρχείο αποθήκευσης των μηνυμάτων. Για τη συμβολοσειρά με την ημερομηνία χρησιμοποιούμε τη συνάρτηση date, η οποία επιστρέφει την τρέχουσα ημερομηνία και ώρα. Η μορφοποίηση αυτών των δεδομένων καθορίζεται από την «παράμετρο μορφοποίησης» που δίνουμε στη συνάρτηση date[2]. Πριν προχωρήσουμε, πρέπει να σταθούμε σε μια λεπτομέρεια: Η ώρα που επιστρέφει η date είναι εκείνη του server και όχι εκείνη του υπολογιστή του χρήστη!

Στη γραμμή 24 προσθέτουμε τα νέα δεδομένα στο αρχείο διατήρησης των μηνυμάτων, χρησιμοποιώντας τη συνάρτηση file_put_contents. Παρατηρήστε ότι για να πετύχουμε την εγγραφή των νέων δεδομένων χωρίς την επικάλυψη των παλιών, δίνουμε και την παράμετρο FILE_APPEND. Επιπλέον, αποθηκεύουμε την τιμή που επιστρέφει η file_put_contents στη μεταβλητή saveGood, για να γνωρίζουμε αργότερα αν η εγγραφή ολοκληρώθηκε με επιτυχία.

```
28 $entries = file(FILE) ;
```

Στη γραμμή 28 χρησιμοποιούμε τη συνάρτηση file και φορτώνουμε όλες τις γραμμές του αρχείου σε έναν πίνακα με το όνομα entries.

```
57  <form name="messageadd" method="post" action="index.php">
58    <label for="nickname">Nickname:</label>
59    <input name="nickname" id="nickname" type="text">
60    <label for="message">Message:</label>
61    <textarea name="message" id="message"> </textarea>
```

2. Μπορείτε να μελετήσετε τη συνάρτηση date και τις τιμές που δέχεται η παράμετρος μορφοποίησης, από την τεκμηρίωση της PHP στη διεύθυνση http://php.net/manual/en/function.date.php

```
62   <input type="submit" value="Place your message">
63 </form>
```

Προφανώς, αυτό το απόσπασμα δεν περιέχει κώδικα PHP. Εδώ βλέπουμε τον κώδικα HTML που δημιουργεί τη φόρμα εισαγωγής μηνυμάτων. Κατά την υποβολή της φόρμας, τα δεδομένα μεταφέρονται με την μέθοδο POST και κατευθύνονται προς το αρχείο index.php – δηλαδή αυτό που εξετάζουμε ήδη! Ακριβώς γι' αυτό, στην αρχή του κώδικα ελέγχουμε αν έχουν φτάσει δεδομένα με τη μέθοδο POST, αν έχει οριστεί το nickname, αν έχει δοθεί κάποιο μήνυμα κ.λπ.

```
66 $errorMsg = "" ;
67 if($savedGood === false)
68   $errorMsg = "Your message hasn't been saved.<br>Please contact the
     webmaster or try again later." ;
69
70 if($emptyForm == true)
71   $errorMsg .= "Please fill your nickname and the message area with
     some text." ;
72
73 if($errorMsg != ""){
74   echo '<ul data-role="listview" data-inset="true">';
75   echo '<li data-role="list-divider">Error</li>';
76   echo "<li> <p> <strong>$errorMsg</strong> </p> </li> </ul>" ;
77 }
```

Ακριβώς κάτω από τη φόρμα θα εμφανίζονται τα μηνύματα της εφαρμογής και, ακολούθως, αυτά των επισκεπτών. Έτσι, στη γραμμή 66 ορίζουμε μια κενή μεταβλητή με όνομα errorMsg και η οποία, όπως πιθανόν μαντεύετε, θα συγκεντρώσει τις περιγραφές από όσα σφάλματα έχουν προκύψει. Στη συνέχεια, στη γραμμή 67 ελέγχουμε αν η μεταβλητή saveGood έχει την τιμή false. Τσεκάρουμε δηλαδή αν έχει συμβεί κάποιο σφάλμα αποθήκευσης ή όχι. Αν ισχύει κάτι τέτοιο, αποδίδουμε ένα σχετικό μήνυμα στην μεταβλητή errorMsg. Στη γραμμή 70 ελέγχουμε αν ο χρήστης υπέβαλε τη φόρμα αφήνοντας κάποιο πεδίο κενό. Σε αυτή την περίπτωση, προσθέτουμε το σχετικό μήνυμα στη μεταβλητή errorMsg. Τελειώνοντας με τους ελέγχους, στη γραμμή 73 τσεκάρουμε αν η μεταβλητή errorMsg είναι κενή ή όχι. Αν όχι, εμφανίζουμε την τιμή της!

```
82           if($entries === false){
83               echo "<li><h3>Ooops!</h3>" ;
84               echo "<p>No guest messages for the moment...</p><p>Be
                 the first to leave one!</p></li>" ;
85           }else{
86               $entries = array_reverse($entries) ;
87               foreach($entries as $entry){
88                   $entryArray = explode("\t", $entry) ;
```

```
89                    echo "<li><p><em>@ $entryArray[0] </em></p>" ;
90                    echo "<h3> $entryArray[1] </h3>" ;
91                    echo "<p><strong>Message:</strong> $entryArray[2]
                      </p></li>" ;
92                }
93            }
```

Σειρά έχει η εμφάνιση των μηνυμάτων των επισκεπτών. Ξεκινάμε (γραμμή 82) με έναν έλεγχο της μεταβλητής entries. Αν αυτή έχει την τιμή false (κάτι που συμβαίνει όταν το αρχείο μηνυμάτων είναι κενό) εμφανίζουμε το σχετικό μήνυμα και μια προτροπή. Αν ο πίνακας έχει περιεχόμενα, φροντίζουμε για την παρουσίασή τους. Αρχικά πραγματοποιούμε μια αντιστροφή της σειράς των στοιχείων του πίνακα, χρησιμοποιώντας τη συνάρτηση array_reverse. Αυτό το κάνουμε γιατί θέλουμε να τα εμφανίσουμε ξεκινώντας από το νεότερο και καταλήγοντας στο παλιότερο[3]. Ακολούθως, επεξεργαζόμαστε τα στοιχεία του πίνακα ένα προς ένα, αξιοποιώντας έναν βρόχο foreach. Η επεξεργασία πραγματοποιείται με τη συνάρτηση explode, η οποία «διασπάει» κάθε στοιχείο του πίνακα (δηλαδή κάθε γραμμή του αρχείου) σε τρία στοιχεία. Αυτή η διάσπαση πραγματοποιείται με χαρακτήρα διαχωρισμού το TAB. Τέλος, στις γραμμές 89 έως 91, προβάλλουμε καθένα από αυτά τα πεδία.

Επίλογος

Είμαστε σίγουροι ότι δεν δυσκολευτήκατε καθόλου να κατανοήσετε τον κώδικα. Επομένως, πιστεύουμε ότι σας αξίζουν συγχαρητήρια! Μόλις κατανοήσατε τον τρόπο λειτουργίας μιας ολοκληρωμένης εφαρμογής σε PHP και, κάτι μας λέει, ότι πλέον είστε σε θέση να φτιάξετε και τις δικές σας. Πριν εμβαθύνουμε τη γνωριμία μας με το περιβάλλον LAMP, μπορείτε να πειραματιστείτε με τον κώδικά μας, να τον τροποποιήσετε ή και να τον αγνοήσετε, ώστε να ξεκινήσετε τη δημιουργία μιας δικής σας εφαρμογής από το μηδέν! Μέχρι το επόμενο τεύχος του περιοδικού, φροντίστε να εξασκηθείτε όσο περισσότερο γίνεται και πάνω απ' όλα να διασκεδάσετε :)

3. Η αλήθεια είναι ότι θα μπορούσαμε να τα παρουσιάσουμε με οποιαδήποτε σειρά, χωρίς να κάνουμε αναδιάταξη των στοιχείων του πίνακα. Στην πραγματικότητα, θέλαμε μόνο να σας δείξουμε τη συνάρτηση array_reverse.

Skill: Intermediate
Tags: wireless, bridging, Man-in-The-Middle, MiTM, Kali

Επιθέσεις Man-in-The-Middle, με wireless bridging

Η δημιουργία ενός wireless access point για επιθέσεις MiTM αποτελεί αγαπημένη δραστηριότητα όσων ασχολούνται με το pen-testing ή έστω απλά ενδιαφέρονται για την ασφάλεια δικτύων. Στο παρόν κείμενο παρουσιάζουμε, αναλυτικά και βήμα προς βήμα, έναν πρωτότυπο τρόπο που μας επιτρέπει να εξαπολύουμε επιθέσεις του είδους εύκολα, γρήγορα κι αποτελεσματικά.



by sub2raw

Κακά τα ψέμματα. Έστω για μία φορά μόνο —και πάντα για καθαρά εκπαιδευτικούς λόγους— όλοι έχουμε ασχοληθεί με το σπορ. Πέρα από τον συνήθη τρόπο για την υλοποίηση μιας επίθεσης MiTM με τη βοήθεια ενός wireless access point (WAP ή απλά AP), υπάρχει κι ένας τρόπος που δεν είναι τόσο δημοφιλής. Το γεγονός αυτό είναι μάλλον παράξενο, όπως θα διαπιστώσετε σε λίγο. Πριν εξηγηθούμε, ας περιγράψουμε το περιβάλλον της εργασίας μας.

Βρισκόμαστε, λοιπόν, στο Kali Linux, έχουμε πλήρη πρόσβαση στο Internet και διαθέτουμε έναν USB WiFi adapter τον οποίο το Kali αναγνωρίζει και συνεργάζεται άψογα μαζί του. Ένας τέτοιος adapter είναι η καρτούλα AWUS 036h της Alfa, την οποία τόσες φορές έχουμε επιστρατεύσει για τους πειραματισμούς μας. Προκειμένου τώρα να προσελκύσουμε wireless clients στο box μας και ν' αρχίσουμε την καταγραφή των δικτυακών πακέτων που στέλνουν και δέχονται (επίθεση Man-in-The-Middle), μπορούμε να ακολουθήσουμε την πεπατημένη. Όποτε το κάνουμε, εργαζόμαστε ως ακολούθως:

- φτιάχνουμε ένα wireless access point
- αποδίδουμε στο WAP μια κατάλληλη για το δίκτυό μας διεύθυνση IP
- στήνουμε έναν DHCP server, ο οποίος μοιράζει διευθύνσεις IP και πληροφορίες DNS στους wireless clients του AP
- εμπλουτίζουμε το routing table του πυρήνα
- ορίζουμε τους κατάλληλους κανόνες για το iptables, ώστε οι clients του AP να επικοινωνούν απρόσκοπτα με μηχανήματα στο Internet
- φροντίζουμε ώστε να είναι ενεργοποιημένο το IP forwarding, μεταξύ των network interfaces.

Υπάρχει όμως κι ένας πολύ πιο σύντομος δρόμος:

- φτιάχνουμε ένα wireless access point
- δημιουργούμε ένα bridge interface που γεφυρώνει δύο ακριβώς network interfaces: εκείνο που αντιστοιχεί στο AP κι εκείνο από το οποίο το Kali βγαίνει στον έξω κόσμο
- αποδίδουμε στο bridge interface μια κατάλληλη, για το δίκτυο στο οποίο ανήκει το Kali, διεύθυνση IP
- φροντίζουμε ώστε να είναι ενεργοποιημένο το IP forwarding, μεταξύ των network interfaces.

Όπως είναι φανερό, η δεύτερη οδός είναι πολύ πιο σύντομη. Το ότι δεν ακολουθείται συχνά, ίσως οφείλεται στο γεγονός ότι το bridging ως έννοια δεν είναι και τόσο διαδεδομένη. Εσείς βέβαια διαβάζετε deltaHacker, οπότε μάλλον έχετε ήδη μάθει όλα όσα χρειαζόσαστε για το bridging (βλ. άρθρο στη σελ. 40). Ας δούμε, λοιπόν, πώς εξαπολύουμε μια επίθεση MiTM, στηριζόμενοι στο λεγόμενο wireless bridging.

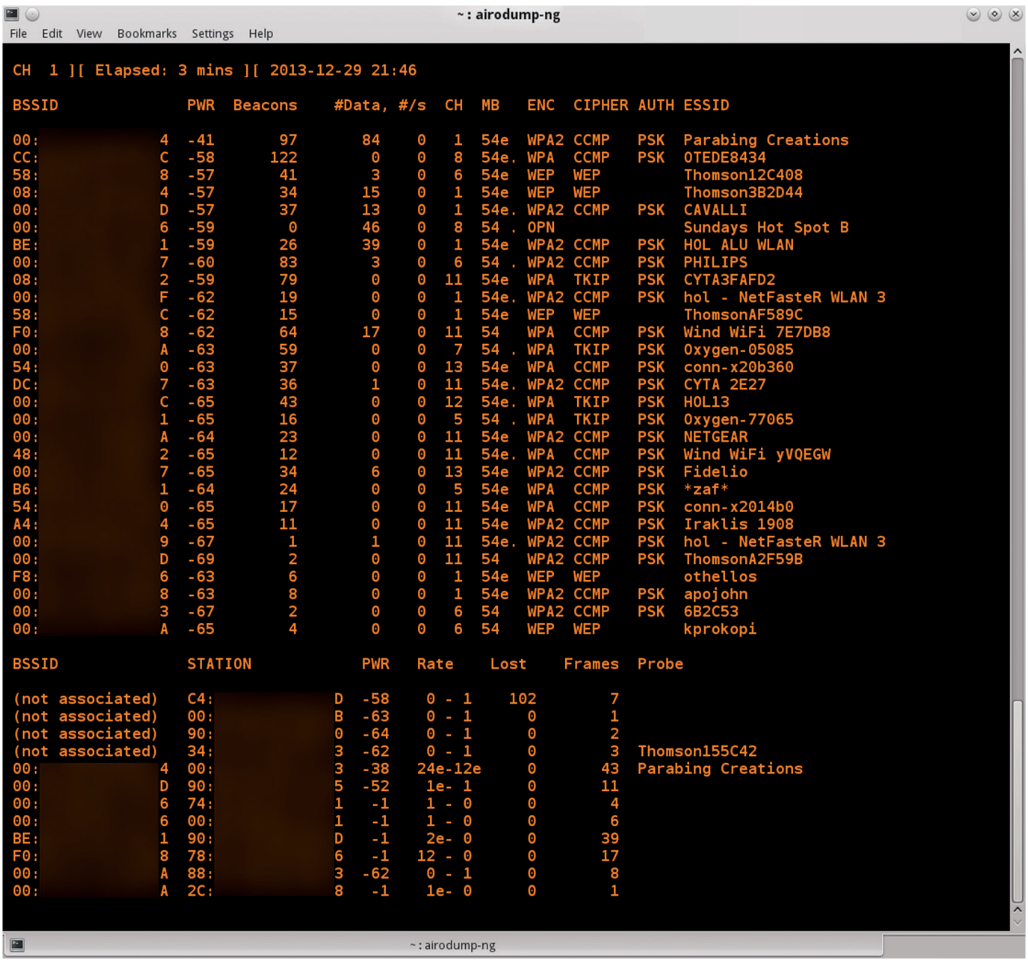
Σημείωση. *Αν κι ο USB adapter που επιστρατεύουμε, προκειμένου να στήσουμε το WAP μας, είναι πέρα για πέρα αληθινός, το Kali Linux είναι εγκατεστημένο σε εικονικό υπολογιστή και συγκεκριμένα σε VirtualBox VM. Σε περίπτωση που θελήσετε να μας μιμηθείτε, ενδεχομένως αυτό το άρθρο να σας φανεί χρήσιμο:* *http://deltahacker.gr/?p=9070*

Βήμα 1

Από ένα παράθυρο τερματικού ελέγχουμε αν ο wireless USB adapter αναγνωρίζεται από το Kali:

```
root@kali64cus:~# airmon-ng
Interface        Chipset          Driver
wlan0            Realtek RTL8187L        rtl8187 - [phy0]
```

Αν την πρώτη φορά που θα δώσετε την παραπάνω εντολή δεν πάρετε τίποτα στην έξοδό της, αποσυνδέστε τον USB adapter, συνδέστε τον ξανά, περιμένετε λίγα δευτερόλεπτα και δώστε την εντολή ξανά.



Ένας τρόπος για να ελέγξουμε ότι δουλεύει σωστά το monitoring interface του Wireless Access Point μας, είναι να ρίξουμε μια ματιά στη γειτονιά. Στη δική μας γίνεται, κλασικά, ο χαμός!

Βήμα 2

Δημιουργούμε ένα monitoring interface:

```
root@kali64cus:~# airmon-ng start wlan0
Found 1 processes that could cause trouble.
If airodump-ng, aireplay-ng or airtun-ng stops working after
a short period of time, you may want to kill (some of) them!
-e
PID     Name
1812    dhclient
Interface           Chipset               Driver
wlan0               Realtek RTL8187L      rtl8187 - [phy0]
                                      (monitor mode enabled on mon0)
```

Αν, όπως στο παράδειγμά μας, δείτε κάποια προειδοποίηση περί του dhclient, αγνοήστε τη. (Αν έχετε ακολουθήσει πιστά τον οδηγό στο `http://deltahacker.gr/?p=9070`, τότε δεν θα δείτε κάποια σχετική προειδοποίηση.)

Βήμα 3

Προαιρετικά, ρίχνουμε μια ματιά στα ασύρματα δίκτυα της περιοχής μας:

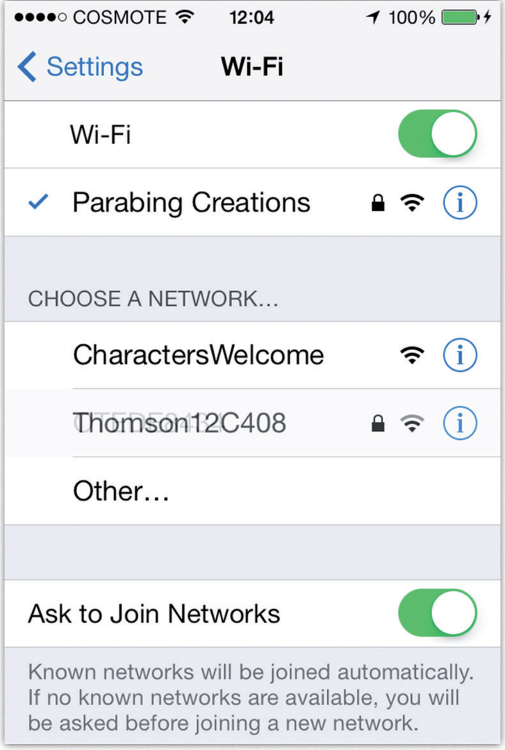
```
root@kali64cus:~# airodump-ng mon0
```

Τερματίζουμε το airodump με τον συνδυασμό πλήκτρων [CTRL+C].

Βήμα 4

Δημιουργούμε κι ενεργοποιούμε ένα wireless access point με όποιο όνομα θέλουμε, το οποίο εκπέμπει στο κανάλι της επιλογής μας:

```
root@kali64cus:~# airbase-ng
--essid CharactersWelcome -c 11
mon0
12:03:40  Created tap interface
at0
12:03:40  Trying to set MTU on at0
to 1500
12:03:40  Trying to set MTU on mon0 to 1800
12:03:40  Access Point with BSSID 00:C0:CA:52:AB:B4 started.
```



Το WAP μας είναι έτοιμο και οι wireless clients το βλέπουν, ωστόσο δεν μπορούν ακόμη να συνδεθούν σ' αυτό.

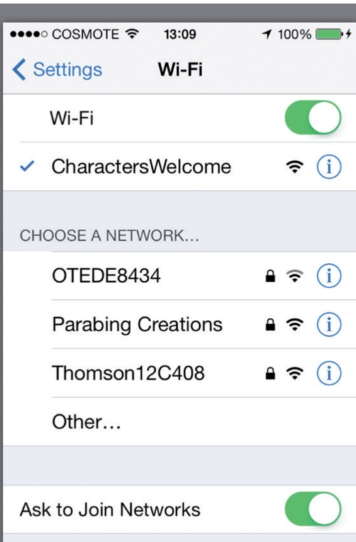
Στο παράδειγμα που μόλις δώσαμε δημιουργήσαμε ένα ασύρματο σημείο πρόσβασης με όνομα CharactersWelcome, το οποίο εκπέμπει στο κανάλι 11. Πλέον, από κάποιον wireless client, όπως, π.χ., είναι ένα laptop ή ένα smartphone,

πηγαίνοντας στο εργαλείο αναζήτησης ασύρματων δικτύων θα πρέπει να βλέπουμε κι αυτό το CharactersWelcome. Βέβαια δεν μπορούμε ακόμα να συνδεθούμε σ' αυτό, είναι όμως εκεί. Από ένα άλλο τερματικό, βλέπουμε το network interface που αντιστοιχεί στο WAP που μόλις ενεργοποιήσαμε:

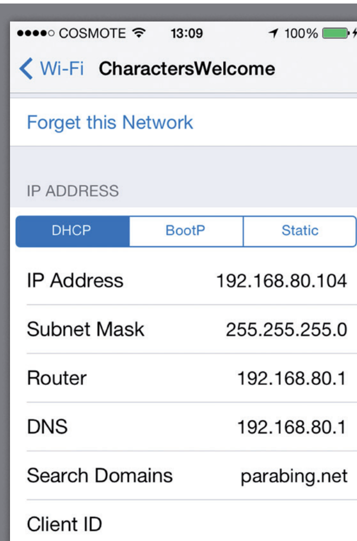
```
root@kali64cus:~# ifconfig -a
at0       Link encap:Ethernet  HWaddr 00:c0:ca:52:ab:b4
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:500
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
eth0      Link encap:Ethernet  HWaddr 08:00:27:5c:32:95
          inet addr:192.168.80.103  Bcast:192.168.80.255
Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe5c:3295/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:43 errors:0 dropped:0 overruns:0 frame:0
          TX packets:21 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000

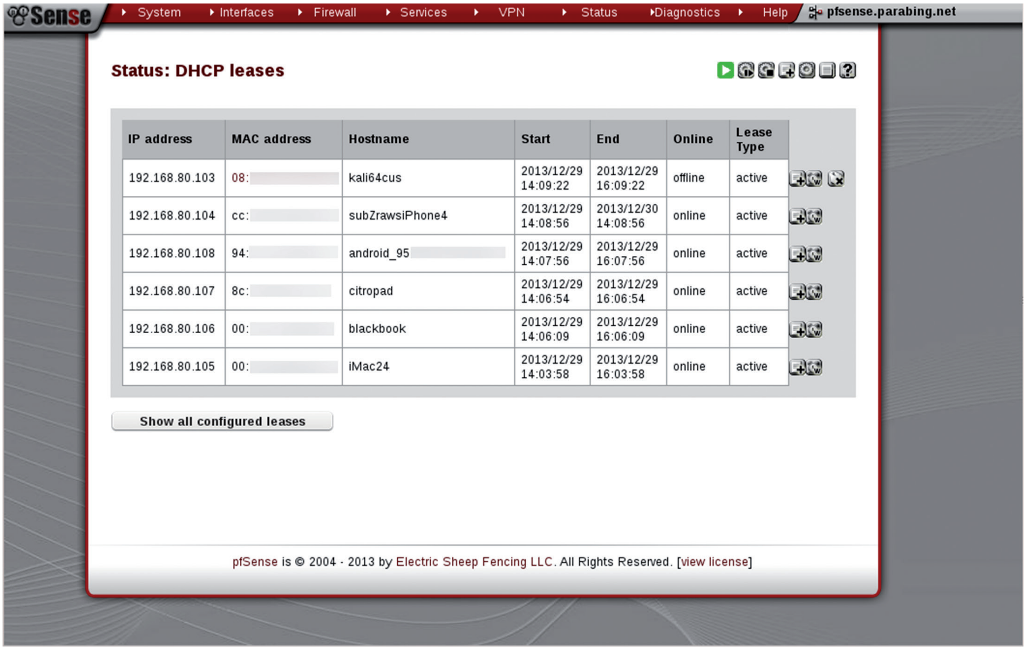
          RX bytes:13262 (12.9 KiB)  TX bytes:8900 (8.6 KiB)
...
```

Το όνομα του interface για το WAP είναι το at0. Κάνουμε και μια νοητή υποσημείωση για το όνομα του network interface από το οποίο το Kali βγαίνει στον έξω κόσμο: Για εμάς ήταν το eth0, κατά πάσα πιθανότητα αυτό θα είναι και για 'σας.



Αφού το network interface του WAP γεφυρωθεί με το eth0 interface του Kali, οι wireless clients μπορούν να συνδέονται άνετα στο ελεύθερο ασύρματο δίκτυο με όνομα CharactersWelcome. Δεν θα το αποτολμούσαν, αν ήξεραν πού πάνε να μπλέξουν...





Οι wireless clients του CharactersWelcome και τα IP που τους έχουν αντιστοιχιστεί δυναμικά, από τον DHCP server στον router μπροστά από το Kali Linux.

Βήμα 5

Ήρθε η ώρα να φτιάξουμε ένα bridge interface που θα γεφυρώνει τα at0 και eth0. Χρειαζόμαστε τα λεγόμενα bridge utilities, τα οποία κατά πάσα πιθανότητα δεν θα είναι εγκατεστημένα:

```
root@kali64cus:~# apt-get install bridge-utils
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  bridge-utils
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/36.1 kB of archives.
After this operation, 146 kB of additional disk space will be used.
Selecting previously unselected package bridge-utils.
(Reading database ... 317241 files and directories currently installed.)
Unpacking bridge-utils (from .../bridge-utils_1.5-6_amd64.deb) ...
Processing triggers for man-db ...
Setting up bridge-utils (1.5-6) ...
```

Φτιάχνουμε τώρα ένα bridge interface, με μέλη τα at0 και eth0:

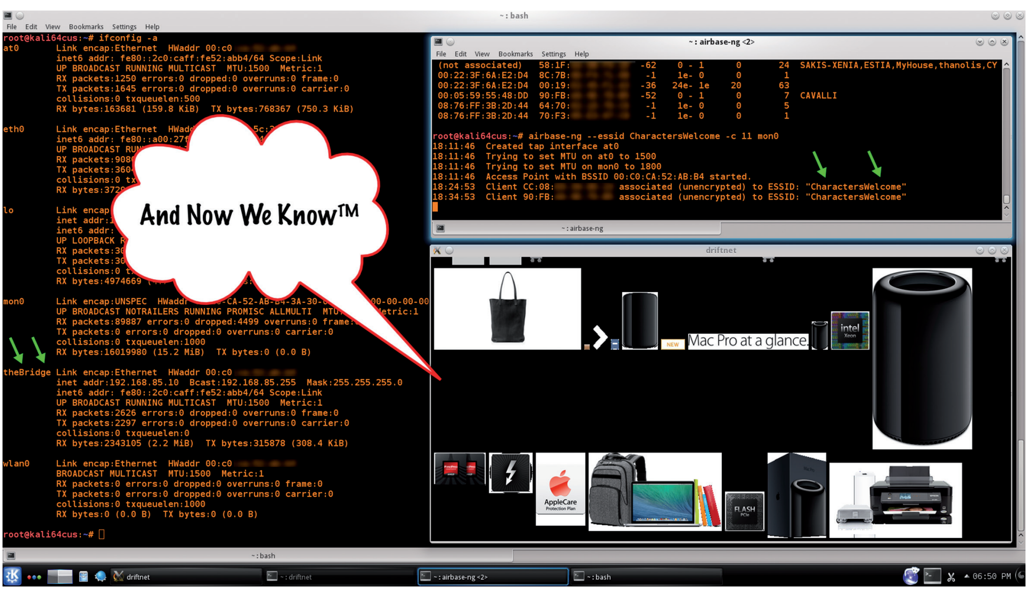
```
root@kali64cus:~# brctl addbr wirelessBridge
root@kali64cus:~# brctl addif wirelessBridge eth0
root@kali64cus:~# brctl addif wirelessBridge at0
root@kali64cus:~# brctl show
bridge name      bridge id             STP enabled    interfaces
wirelessBridge   8000.00c0ca52abb4     no             at0
                                                      eth0
```

Παρατηρήστε ότι το δικό μας bridge interface το ονομάσαμε wirelessBridge. Φροντίζουμε ακολούθως ώστε τα at0 και eth0 να είναι ενεργά, αλλά χωρίς διευθύνσεις IP:

```
root@kali64cus:~# ifconfig eth0 0.0.0.0 up
root@kali64cus:~# ifconfig at0 0.0.0.0 up
```

Δίνουμε στο bridge interface μια διεύθυνση IP κατάλληλη για το LAN μέσα στο οποίο βρίσκεται το Kali. Στην περίπτωση του δικού μας δικτύου, ο DHCP server στον router μπροστά από το Kali μοίραζε διευθύνσεις από την περιοχή 192.168.80.100-150. Επιλέξαμε να αντιστοιχίσουμε χειροκίνητα μια στατική διεύθυνση IP στο wirelessBridge, η οποία ήταν εκτός του προαναφερθέντος εύρους και δεν χρησιμοποιούταν από κανέναν άλλον υπολογιστή.

```
root@kali64cus:~# ifconfig wirelessBridge 192.168.80.10/24 up
root@kali64cus:~# ifconfig wirelessBridge
wirelessBridge Link encap:Ethernet  HWaddr 00:c0:ca:52:ab:b4
     inet addr:192.168.80.10  Bcast:192.168.80.255  Mask:255.255.255.0
     inet6 addr: fe80::2c0:caff:fe52:abb4/64 Scope:Link
```



Η επίθεση MiTM είναι σε εξέλιξη κι ο attacker χρησιμοποιεί το εργαλείο driftnet, προκειμένου να βλέπει και να αποθηκεύει τις εικόνες στα sites που επισκέπτονται τα ανυποψίαστα θύματά του.

```
UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
RX packets:6 errors:0 dropped:0 overruns:0 frame:0
X packets:6 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:1446 (1.4 KiB)  TX bytes:468 (468.0 B)
```

Βήμα 6

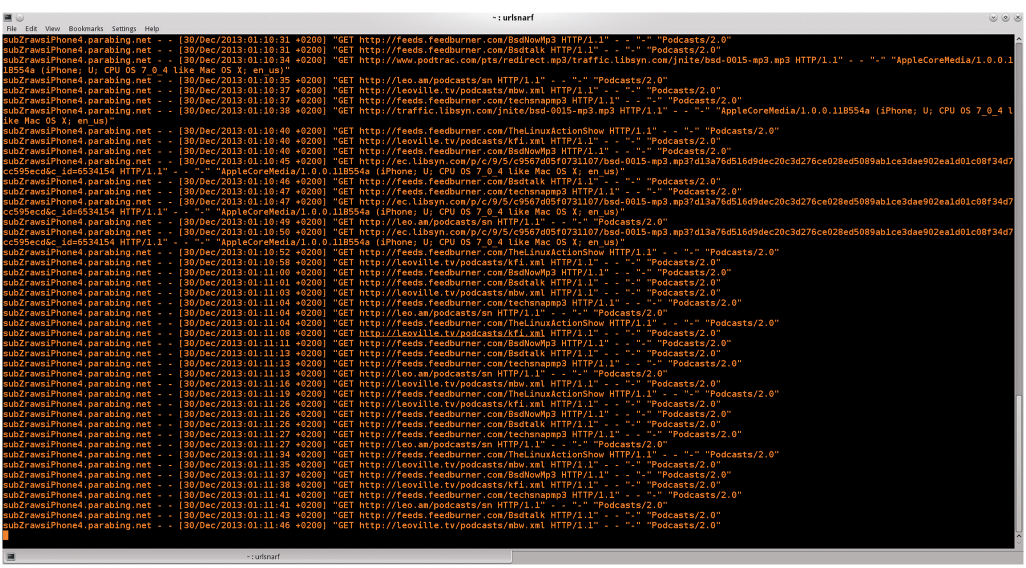
Τελευταίο βήμα είναι η ενεργοποίηση του IP forwarding μεταξύ των network interfaces:

```
root@kali64cus:~# echo 1 > /proc/sys/net/ipv4/ip_forward
```

Δοκιμές και δυνατότητες

Δοκιμάστε να συνδεθείτε στο ασύρματο δίκτυο ονόματι CharactersWelcome, από έναν οποιονδήποτε wireless client. Αν όλα έχουν πάει καλά η σύνδεση θα επιτευχθεί χωρίς προβλήματα και χωρίς φυσικά να ζητηθεί κάποιο password (έχουμε φτιάξει ένα ανοικτό ασύρματο δίκτυο). Ο client θα πάρει διεύθυνση IP από τον DHCP server μπροστά από το Kali (και στο πλαίσιο του παραδείγματός μας, από το εύρος διευθύνσεων 192.168.80.100-150).

Από τη στιγμή που όλη η δικτυακή κίνηση των wireless clients του CharactersWelcome περνά από τα network interfaces του Kali, έχουμε ουσιαστικά πετύχει μια επίθεση MiTM εναντίον τους. Πέρα από το κλασικό, νομίζουμε, Wireshark, κατά τη διάρκεια των πειραματισμών σας δοκιμάστε τα driftnet και urlsnarf. Δείτε τα σχετικά screemshots με τις περιγραφές τους, αλλά κυρίως πειραματιστείτε. Καλή σας διασκέδαση!



Σε ένα άλλο τερματικό ο attacker τρέχει το urlsnarf, ώστε να βλέπει τις διευθύνσεις (URLs) που επισκέπτονται οι clients του ελεύθερου WAP που έχει στήσει.